8-2009

Geometric Build-up Solutions for Protein Determination via Distance Geometry

Robert Tucker Davis
Western Kentucky University

GEOMETRIC BUILD-UP SOLUTIONS FOR PROTEIN DETERMINATION VIA DISTANCE GEOMETRY

A Thesis
Presented to
The Faculty of the Department of Mathematics
Western Kentucky University
Bowling Green, Kentucky

In Partial Fulfillment
Of the Requirements for the Degree
Master of Science in Applied Mathematics

By
Robert Tucker Davis

August 2009

GEOMETRIC BUILD-UP SOLUTIONS
FOR
PROTEIN DETERMINATION VIA DISTANCE GEOMETRY

Date Recommended_____

_____
Dr. Di Wu

_____
Dr. Claus Ernst

_____
Dr. Claire A. Rinehart

_____
Dean, Graduate Studies and Research Date

Acknowledgements

It is with the utmost gratitude that I acknowledge those people in my life who have not only contributed to and guided the direction of this thesis, but who have also encouraged me and shaped me as a person. It would be quite impossible for me to acknowledge and fully thank everyone I would like to and brevity is called for.

I would first like to thank my thesis advisor, Dr. Di Wu, who, in the beginning, told me of the bio-mathematics of protein folding and the molecular distance geometry problem. I really liked that it was such an inter-disciplinary area and asked him if I would be able to work with him. Initially, Dr. Wu showed great faith in me by taking me on as his advisee facing such a difficult problem, as I had little to no experience in Biology, Biochemistry, or Computer Programming. Through our work together, he has proven himself to be one of the most knowledgeable people I have ever had the pleasure of meeting. He has also advised me with much patience throughout the learning process and a firm hand, when needed. He motivated me with encouragement rather than fear; and guided me with general directions allowing me to find my own path. In short, thank you Dr. Wu for all that you've done, for the trust and time that you have spent on my behalf.

I would also like to thank the other members of my committee, Dr. Claus Ernst and Dr. Claire Rinehart. These gentlemen have contributed many hours to reviewing and revising this thesis as well as offering advice. Dr. Rinehart has shown great patience with my lack of knowledge regarding general Biochemistry and, further, has been a willing and skillful teacher. Dr. Ernst has taught me in topology and has now, seemingly,

become my colleague. He has influenced me regarding my career and education. Thanks Dr. Rinehart for the friendly help and my friend, Claus, thanks for everything.

I would also like to thank everyone in the Mathematics Dept. at WKU who has helped me with their kind words and invaluable instruction. My tenure here has been long and shaped by a great many subtle influences. Thank you all.

Lastly, with all my heart, I would also like to thank my family. My parents have provided my brother and me with all of life's necessities, with love and support. They played an invaluable role in my ability and opportunity to attend WKU. What's more important, though, was their willingness, even strange desire, to listen to me rant about specific problems, even when they did not know what I was talking about. They and my brother endured the vocalizations and manifestations of my stress and frustrations. For this I am sorry, but forever grateful. My brother has not only listened and supported, but has also loaned me his vehicle and given me rides while I was without. In many things and in many ways whatever I lack or whenever I am not able, he has or is able. Mom, Dad, Luke: thank you, thank you, and thank you.

Table of Contents

Table of Contents (Continued)

GEOMETRIC BUILD-UP SOLUTIONS
FOR
PROTEIN DETERMINATION VIA DISTANCE GEOMETRY

Robert Tucker Davis August 2009 65 Pages

Directed by: Di Wu[1], Claus Ernst[1], and Claire A. Rinehart[2]

1. Department of Mathematics Western Kentucky University
2. Department of Biology Western Kentucky University

Proteins carry out an almost innumerable amount of biological processes that are absolutely necessary to life and as a result proteins and their structures are very often the objects of study in research. As such, this thesis will begin with a description of protein function and structure, followed by brief discussions of the two major experimental structure determination methods. Another problem that often arises in molecular modeling is referred to as the Molecular Distance Geometry Problem (MDGP). This problem seeks to find coordinates for the atoms of a protein or molecule when given only a set of pair-wise distances between atoms. To introduce the complexities of the MDGP we begin at its origins in distance geometry and progress to the specific sub-problems and some of the solutions that have been developed. This is all in preparation for a discussion of what is known as the Geometric Build-up (GBU) Solution. This solution has lead to the development of several algorithms and continues to be modified to account for more and different complexities. The culmination of this thesis, then, is a new algorithm, the Revised Updated Geometric Build-up, that is faster than previous GBU's while maintaining the accuracy of the resulting structure.

Chapter 1: Introduction

1.1 Motivation and Research Goals

Biology is the study of living organisms, and biologists focus much of their attention on the different types of cells that make up an organism. The different types of cells have their own functions, structures and properties that are predominantly determined by proteins. As a result, proteins themselves are very often the object of study. They are one of the largest types of biological macromolecules containing on average around 6650 atoms compared to lipids, another type of macromolecule, which have on average only 95 atoms. Furthermore a protein's amino acid sequence determines a three-dimensional structure which is unique to that protein. The atomic structure of a protein defines areas called binding sites which are sometimes realized as a depression in the surface of the protein and more often realized as a patch on the surface. It is these binding sites, created by the three-dimensional atomic structure of a protein, that act like a key fitting a keyhole on another molecule, which determines its function. Ultimately then, knowing a protein's structure can give clues or indicators of its function. Protein Structural determination is the process of "solving" a protein's three-dimensional atomic structure by finding 3D coordinates for all of the atoms in the protein.

To determine the three-dimensional structure of proteins, biologists and physicists have developed many different methods to extract structural data, including X-ray crystallography and Nuclear Magnetic Resonance (NMR). Inter-atomic distances, bond lengths, bond angles, and dihedral angles are just some of the structural data extracted and it is from these data a protein structure can be determined computationally and it is generally formulated as a so called Molecular Distance Geometry Problem (MDGP).

The MDGP is itself split into three sub-problems: all exact distances, sparse exact distances, and distance ranges. In 2002 Dong and Wu first described the geometric build-up solution as a linear time algorithm for solving the molecular distance geometry problem [1]. This work was then extended to solve the case of sparse exact distances [2], while also minimizing the total error [3]. In the two latter cases the algorithm's running time is severely affected by the search for a set of base atoms which possess all of the required distances.

The goal of this thesis is to present a new addition to the family of geometric build-up solutions, a Revised Updated Geometric Build-up algorithm. This algorithm has been designed to include data structures as well as a triangle detection method in an attempt to reduce the running time of the geometric build up solution. In addition to this, the proposed revised updated geometric build-up algorithm also employs the updating routine, first described by Wu and Wu, as a means of minimizing the total error of the structure resulting from computational round-off error.

1.2 Thesis Outline

This thesis will explore the union of the Distance Geometry Problem and protein determination as well as a survey of some of the existing algorithms from a family of geometric build-up (GBU) solutions and give a new GBU algorithm for solving the Molecular Distance Geometry Problem (MDGP). Specifically, Chapter 2 is devoted to an overview of proteins beginning from RNA transcription. A description of protein structural composition and its relationship to protein functions are described briefly in section 2.1. Section 2.2 goes on to describe the two major experimental methods (x-ray

crystallography and NMR spectroscopy) that biologists use to determine the three-dimensional structure of proteins. This section emphasizes the scientific basis of each method as well as some pros and cons of each.

Chapter 3, while including some distance geometry history, focuses on the general Distance Geometry Problem (DGP), and its application to the area of molecular modeling and protein determination known, as the Molecular Distance Geometry Problem (MDGP). Section 3.1 describes the *k*-dimensional distance geometry problem including the relative difficulty of the problem and a short discussion of the theory of NP-completeness. The molecular distance geometry problem and the DGP limited to three-dimensions, are described in section 3.2. This section goes on to formulate the MDGP as three sub-problems: all exact, sparse exact, and distance ranges. In describing the all exact and distance range cases, existing solutions are described along with their relative properties.

Dong and Wu first described the geometric build-up (GBU) solution [5], which has yielded a family of algorithms. This thesis extends this family of GBU algorithms by provided solutions to the sub-problems of sparse exact distances. Chapter 4 begins with a survey of the geometric build-up algorithms starting with the basic idea (section 4.1), then extends to the case where all exact distances are available (section 4.2). This survey presents the algorithms in a manner that mirrors the natural progression or evolution of the geometric build-up solutions. Section 4.3 discusses the solution's application to the sparse case of the MDGP pointing out the dilemma of error propagation. Section 4.4 then describes an updating routine and algorithm that enables the GBU solution to control and minimize this error. Finally, the Revised Updated Geometric Build-up Algorithm

(RUGB) is offered as the main result of this thesis in section 4.5. This section includes test results that illustrate the run time and accuracy of the revised updated GBU algorithm.

This thesis concludes, in Chapter 5, with a summary of the research project and computational results. Also included in the conclusion are future directions of study and extensions of this work.

Chapter 2: Introduction to Protein Structure

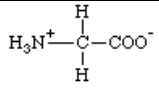
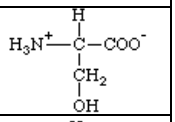
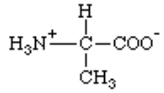
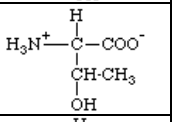
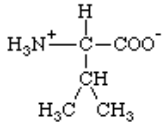
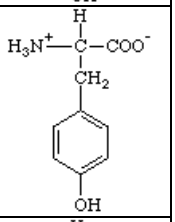
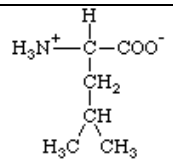
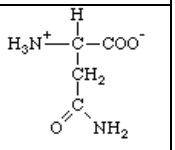
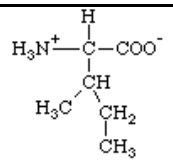
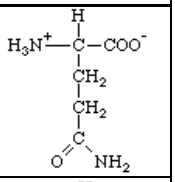
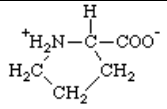
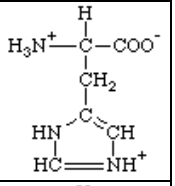
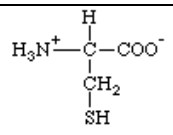
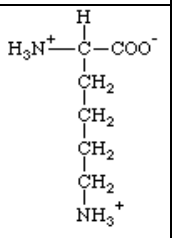
2.1 Structural Biology of Proteins

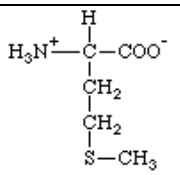
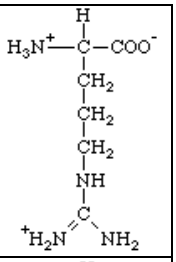
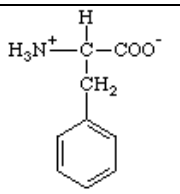
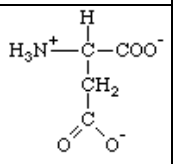
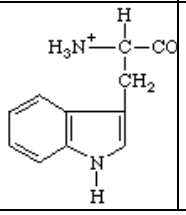
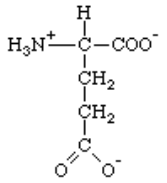
Proteins are molecules with important functions in the biological activities of life. They were first discovered in the 1830's by Jons Jakob Berzelius and his advisee Johannes Mulder who coined the term proteins [4] after the Greek word "prota" which means "of primary importance". A couple of factors contributed to this name, not the least of which was the fact that Berzelius noticed that plants prepared proteins as the primary source of animal nutrition. Today proteins are known to no only contribute to nutrition but to also serve as biological catalysts, regulatory sensors and structural building blocks required in the process of life.

To further understand the three-dimensional structure of proteins it is important to begin at synthesis and understand how they are made and what they are made of. In the 1870's Heinrech Hlasiwetz and Josef Habermann discovered that the building blocks of proteins are amino acids [5]. All proteins are made from the same set of 20 amino acids and differ only in their order and composition. Amino acids have common structure based around a central alpha-carbon atom. The alpha-carbon is bonded to an amino group (NH_3^+), a carboxyl group (COO^-), a hydrogen atom (H) and an R group known as a side chain. The side chain is the defining characteristic of the amino acids because it is responsible for the varying properties of them [6][7]. Table 2.1 lists the 20 amino acids and their chemical skeletal structures. It is important to know that these amino acids are not planar and instead the amino, carboxyl, R groups

along with the hydrogen atom are arranged spatially around the alpha-carbon atom in the form of a tetrahedron [6][7].

Table 2.1: List of 20 amino acids

Non-Polar			**Polar**		
Name	**Abbr.**	**Structure**	**Name**	**Abbr.**	**Structure**
Glycine	G		Serine	S	
Alanine	A		Threonine	T	
Valine	V		Tyrosine	Y	
Leucine	L		Asparagine	N	
Isoleucine	I		Glutamine	Q	
Proline	P		Histidine (Basic)	H	
Cysteine	C		Lysine (Basic)	K	

Methionine	M		Arginine (Basic)	R	
Phenylalanine	F		Aspartic Acid	D	
Tryptophan	W		Glutamic Acid	E	

Frederick Sanger showed that the amino acids that make up a protein begin as a linear sequence [8]. The construction of the sequence follows what is known as the "Central Rule" which states that each gene in the DNA contains the "blueprint" for a single protein's sequence. First the gene must be transcribed as messenger RNA (mRNA), which is a sequence of nucleotides (Figure 2.1.A). These nucleotides are read as triplets called codons and each codon represents a specific amino acid (Figure 2.1.B). The required amino acids are carried to the assembly site on ribosomes by transfer RNA (tRNA) which is specific to the amino acid and to the codon (Figure 2.1.C). Once a new amino acid-tRNA complex is in place on the ribosome it is then joined in a linear sequence, to the growing protein by a peptide bond between the carboxyl group, from one amino acid, to the amino group of the second amino acid (Figure 2.1.D)[6][7]. These

sequences, known as the protein primary structure, usually average between 200-300 amino acids [6].

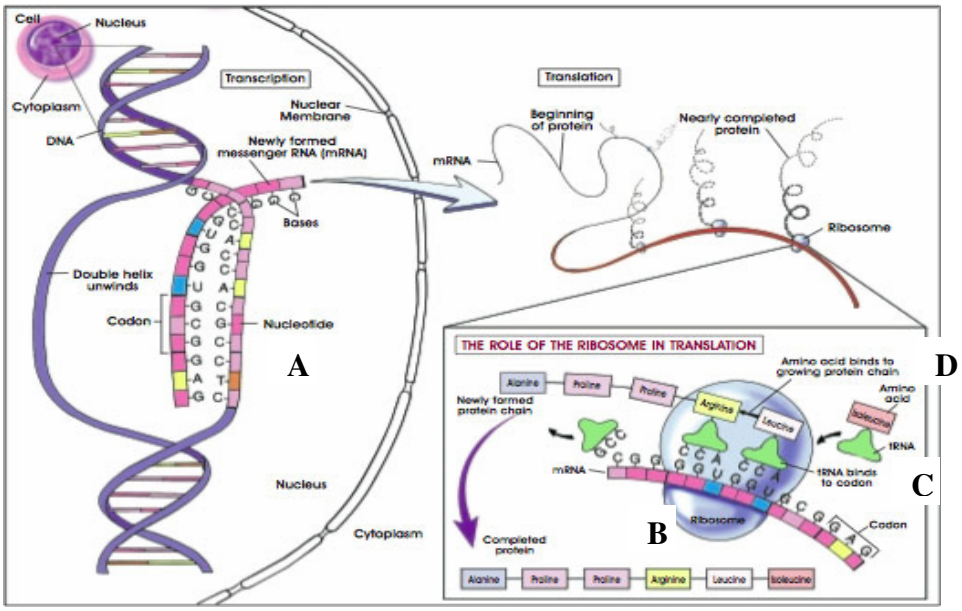


Figure 2.1: Illustration of the central rule and protein synthesis
Borrowed from National Institutes of Health

This amino acid sequence was a major discovery in protein studies. It was Christian Anifinsen who showed that the sequence is primarily responsible for the protein's three-dimensional structure [9] because of certain properties inherent to the atoms that make up the amino acids in the sequence. After the assembly the amino acids in the chain begin to react with one another and to their environment in a process known as protein folding [6]. Because the peptide bonds are formed at the amino and carboxyl groups the hydrogen atom and the side chain are the only protrusions from the backbone. Hydrophobic and polar interactions between the sidechains and their environment are two of the main driving forces behind protein folding. Table 2.1 classifies the amino acids into polar and non-polar. Polar amino acids readily interact with water and are therefore soluble in an aqueous environment. Non-polar amino acids form hydrophobic

interactions with each other and cell membranes. These interactions display the importance of the environment on the protein and its location in the body, which can also determine the activity or inactivity of the protein.

After the initial positioning of the amino acid R-groups into their respective polar and non-polar environments, weak hydrogen bonds form local three-dimensional structures known as alpha-helices and pleated beta-sheets. These local structures are referred to as the secondary structures of a protein and together with turns in the sequence comprise the entire three-dimensional native structure of the protein known as its tertiary structure. Other major contributors to the stabilization of protein three-dimensional structure include disulfide bridges which are covalent bonds between the sulfur atoms of non-adjacent Cysteine side-chains, and ionic interactions the positive (+) and negative (-) charges of different side-chains. Other, weaker, forces also contribute to the folding process including van der Waals interactions and ring stacking.

The first to determine or solve the three-dimensional structure of any protein were Max Perutz and John Cowdery Kendrew who solved the structures of Hemoglobin [10] and Myoglobin, respectively [11]. Myoglobin's three-dimensional structure consists of a single chain of amino acid residues, when this is the case the three-dimensional stable conformation is known as the protein's tertiary structure. Very often, as in hemoglobin, a protein consists of multiple amino acid chains held together by the stabilizing forces described above. This complex of chains is called the proteins quaternary structure [6].

Proteins have a wide variety of functions and certainly many of the processes in life would not take place without them. These almost endless functions can be placed into two categories, structural and enzymatic [6][7]. Some proteins are ligand binding

like hemoglobin which attaches to oxygen in the lungs and shuttles it to the extremities of the body where it releases it to oxygenate the tissue. Similarly, proteins known as antibodies bind to foreign contaminants in the body so they may be destroyed. Other proteins are enzymatic in that they are responsible for catalyzing reactions in the cell such as those involved in metabolism, DNA replication, DNA repair, and transcription. Signaling proteins are responsible for carrying messages from cell to cell and membrane proteins bind to them as receptors. Fibrous proteins are responsible for cell structure and elasticity like collagen. Motor proteins like actin are responsible for cell motility and muscular contractions. These are but a few examples of the many functions proteins are responsible for and, as this is a growing area of study, more and more is being discovered about their functions. For more on protein functions see the referenced textbook [6][7].

To summarize, the amino acid sequence uniquely determines a protein's three-dimensional structure, which in turn determines the function and utility of the protein. Before it is functional, a protein must undergo proper folding. Incorrectly folded proteins are broken down and the amino acids are reused. If an incorrectly folded protein cannot be broken down, it may be discarded and left to form aggregates. Localization of such aggregates can cause Mad Cow disease and other degenerative brain diseases. Understanding the role that proteins play in disease continues to be a major motivating factor for the study of proteins. Currently one of the major areas of protein studies is structure determination which leads to a more thorough understanding of just how these proteins carry out their functions.

2.2 Main Methods of Protein Structure Determination

Protein structures are determined by two major experimental methods X-ray crystallography and Nuclear Magnetic Resonance (NMR) Spectroscopy. From statistical information of the Protein Data Bank (PDB) online, which catalogues all proteins that have been determined, as of May 31, 2009, the PDB shows that of the 53,435 proteins that are catalogued 46,296 (86.6%) have been determined using X-ray crystallography while only 6,852 (12.8%) proteins have been determined using NMR Spectroscopy.

The process of X-ray crystallography begins by isolating a protein in a sodium chloride crystal, which was first accomplished by James B. Sumner [12]. The crystal is then placed into a diffractometer or goniometer where x-rays are blasted through the crystal at the protein. The regular pattern of the crystal allows the x-rays to diffract off of the electons of each atom. This means the x-rays split into multiple beams with different directions that are used to create two-dimensional diffraction images. These images and other spatial data are then converted to an electron density map and subsequently to a single three-dimensional image. The process of converting the data and 2-D images to a 3-D structure is actually quite computing intensive and requires the use of Fourier Transforms.

This method has proven to be the most useful; however it is not without its drawbacks. Although the process of growing the crystal has been streamlined since its inception, it can sometimes be very time consuming and some proteins such as membrane proteins cannot be easily crystallized, if at all. One of the limitations in crystal growing is the requirement for large quantities of pure protein. Moreover, the crystal must be of a certain quality; for instance if the crystal is too small or has imperfections the resolutions

may be off producing incorrect results. For this reason most biologists grow multiple crystals with the hope that one of them will be of a sufficient quality.

The theory of Nuclear Magnetic Resonance (NMR) was first observed by Isidor Rabi in 1938. What Rabi observed, that was later described by Felix Bloch and Edward Purcell, was that elements with an odd number of nucleons (protons and electrons) have a well known spin and that the spin enables it to be affected by a magnetic field. They found that those specific atoms can absorb radio frequencies when exposed to a magnetic field [13][14]. The frequency at which absorption of the radio wave occurs is unique to coupling between local atoms and thus by varying the radio frequency used, the scientist can observe spin shifts of atoms to gather physical, chemical, electronic, and structural data. This spectroscopy process begins by first placing a protein into an NMR transparent solution which is then placed into a spectrometer where a constant magnetic field and varying radio frequencies can be applied and the data collected. The data is then used to construct the protein's three-dimensional structure, and it is the structural data that will be of most relevance to this thesis; specifically inter-atomic distance data.

This method too has drawbacks, however. One of the major inhibitors is that not all atoms are observable; that is, only isotopes with odd isotope numbers. These are generally not well represented in proteins and therefore if NMR is to be successful they must be enriched to a sufficient number so enough data can be collected. Another drawback of the NMR is that the proteins are placed into a solution. Due to their solubility, the atoms will have a slight wiggle which yields inter-atomic distance ranges rather than exact distances. As such NMR spectroscopy data may result in multiple structures satisfying the distance constraints. Yet another downside to NMR is that it will

currently only work for small proteins due to the crowding of the 1-dimensional signal spectrum. To get around this problem and to enable the NMR analysis of larger proteins some multidimensional (2D, 3D, and 4D) methods have been developed [15].

These two methods are only the major experimental methods employed; many alternative methods exist such as cryo-electron microscopy, fiber diffraction, mass spectrometry, circular dichroism, etc. There also exist some theoretical methods such as Potential Energy Minimization, which is sometimes used to refine an experimentally determined function [16]. All of these methods offer their own pros and cons, but all methods represent attempts to determine structural data of a molecule. All of the major methods discussed require advanced mathematical and physical methods to construct the protein's structure from the experimental or theoretical data. X-ray crystallography and NMR both use Fourier transforms to realize the three-dimensional structure from the data and potential energy minimization creates a potential energy function, which can quickly become very complex, and then requires the solving of a multidimensional optimization problem.

Chapter 3: Origins and Formulation of the Problem

3.1 The Distance Geometry Problem

Inter-atomic distances of a protein structure can often be obtained experimentally or theoretically, then the protein 3D structure can be determined. However, it requires solving a challenging mathematical problem called the distance geometry problem. Distance Geometry began in 1841 when Arthur Cayley was the first to state the general form of the distance geometry problem [17], but it was Karl Menger, in 1923, who established distance geometry as its own branch of mathematics by showing that many geometric properties could be formulated and examined using only pair-wise distances between points [18]. In distance geometry we begin with an object that is defined only by a subset of distances. The distance geometry problem (DGP) then is to compute the coordinates for each point so that the given distances are satisfied [17][19]. In the application of distance geometry, biomolecular modeling, three-dimensional models of proteins structures are generally considered. L.M. Blumenthal, stated the problem as, "When we have a given set of distances between pairs of points, the distance geometry can give a clue to find a correct set of coordinates for the points in three-dimensional Euclidean space satisfying the given distance constraints" [19]. This is often referred to as the Molecular Distance Geometry Problem (MDGP). The MDGP is itself divided into three sub-problems, which will be formalized later. The general case of the DGP has been shown to be NP-hard, prompting the following description of the computational complexity of the DGP.

3.1.1: Introduction to the theory of NP

In 1979, J.B. Saxe showed that the DGP belongs to a class of problems known as NP [20]. This classification arose as a way to measure a problem's difficulty. Formally, NP is the class of decision problems that are solvable by a Nondeterministic Turing Machine in polynomial time [21] or equivalently problems that are verifiable by a Deterministic Turing Machine in polynomial time [22]. A Turing Machine is a theoretical machine which performs an action and moves to another input based upon a state and a current input. If the Turing Machine is deterministic (DTM) a single move is specified and if the machine is non-deterministic (NTM) then multiple moves may be possible based upon the state and the single input.

This is measured by the "size" of the problem, which is the amount of input data required to explicitly describe the problem instance [23]. For the DGP the size would be the number of points, because coordinates for each point must be found to describe them all as a single instance. The time complexity for these problems is given as an upper bound on the amount time required to solve a problem of that particular size. To measure this, a method known as "Big O" notation has been developed so to ignore all variability between computing machines and measure the time required to solve only in terms of the problems size. Mathematically,

$$f(n) = O(g(n)) \text{ iff } \exists\, c \in R^{+},\ n,k \in N \text{ such that } \forall\, n \geq k,\ \left|f(n)\right| \leq c \cdot \left|g(n)\right|.$$

Thus the formal definition of a polynomial time algorithm is one whose complexity function is $O(p(n))$ for some polynomial function p given in terms of the problem size n [23]. If an algorithm cannot be bounded by a polynomial function then the problem is

said to be an exponential time algorithm regardless of being a true exponential function. These complexity classes are now more formally defined [24].

Definition 3.1.1: NP is the complexity class of decision problems for which answers can be checked by an algorithm whose run time is polynomial in the size of the input.

Definition 3.1.2: P is complexity class of languages that can be accepted by a deterministic Turing machine in polynomial time.

Definition 3.1.3: NP-hard is the complexity class of decision problems that are intrinsically harder than those that can be solved by a nondeterministic Turing machine in polynomial time.

Definition 3.1.4: Strongly NP-hard is the complexity class of decision problems which are still NP-hard even when all numbers in the input are bounded by some polynomial in the length of the input.

Definition 3.1.5: NP-complete is the complexity class of decision problems for which answers can be checked for correctness, given a certificate, by an algorithm whose run time is polynomial in the size of the input (that is, it is NP) and no other NP problem is more that a polynomial factor harder.

The idea of NP extends from decision problems to optimization problems and the details of the Turing Machines are ignored as the class, NP, is applied to algorithmic solutions verifiability. Informally, if every instance of a problem can be solved by a polynomial time algorithm, the problem belongs in the class P. This class is a subset of the class NP, in which not every instance of a problem, if any, will be solvable by a

polynomial time algorithm. Also included under the NP umbrella class are those recognized as the most difficult problems, the class NP-complete.

3.2 The Molecular Distance Geometry Problem

The application of the distance geometry problem is known as the Molecular Distance Geometry Problem (MDGP). Crippen and Havel were the first to apply the DGP to the area of molecular modeling, using experimentally obtained (X-ray or NMR) distance data [17]. They developed the EMBED algorithm that will be discussed later and this algorithm is a very important work in the area of distance based molecular modeling. Other existing solutions include a graph reduction software package ABBIE, developed by Hendrickson [25]; the Alternating Projection Algorithm by Glunt et al [26] and DGSOL by More and Wu.

In practice this is a rather difficult problem, for a few major reasons. The largest impediment to this problem is that only a small subset of the total inter-atomic distances can be obtained. With fewer distances the problems difficulty increases and at least one solution, the Singular Value Decomposition, requires a full set of distances to be able to find coordinates for all atoms in a protein. This solution will be discussed in more detail later. Another downside, mentioned in the description of the DGP, is that the distances collected often contain inconsistencies or errors that violate geometric properties like the triangle inequality. For example, if $d_{i,j}$ is the distance between the points i and j, and similarly we have $d_{i,h}$ and $d_{j,h}$, then $d_{i,j} \leq d_{i,h} + d_{j,h}$. If this is the case then it will not be at all possible to find a complete set of coordinates satisfying the given constraints unless a small amount of error is allowed [27]. Still another setback of the practical case is that

the distance data experimentally obtained through NMR spectroscopy are distance ranges, not exact distances. For this reason an ensemble of possible structures may be found satisfying the constraints at which time the most accurate would then have to be located from that family of structures.

For these reasons the MDGP is classified into three sub-problems, first is when a complete set of exact distances are known, second is when only a sparse set of exact distances, and the third type is when instead of having exact distances only lower and upper bounds, or constraints, on the distances are known. The latter is the most practical of the three sub-problems but the first two are also important because their solution can lead to a greater understanding of the real-world situation as well as important improvements in the methods of solution. This is the goal of this thesis: to find an efficient algorithm for the case of sparse exact distances with the hopes of later extending its applications to the more practical case of sparse distance ranges. Here the MDGP sub-problems will be formally defined and followed by a brief discussion of each.

3.2.1 MDGP with All Exact Distances

In the case where a complete set of exact distances are available the problem is formulated as follows.

MDGP: All Exact Distances

Let $a_1, a_2 \ldots a_n$ be the atoms of a protein and $d_{i,j}$, be the inter-atomic distances for complete set S of the pairs of atoms (a_i, a_j) $1 \leq i, j \leq n$. The problem then is to find

coordinates in 3-dimensions $x_i = (x_{i,1}, x_{i,2}, x_{i,3})$ such that the distances between pairs of points x_i and x_j, given by the Euclidean norm (||.||), will satisfy the given distances.

$$\left\| x_i - x_j \right\| = d_{i,j} \quad (i, j) \subseteq S \tag{1}$$

One of the earliest solutions to this problem utilizes the Singular Value Decomposition of an induced or modified distance matrix [17][19]. Before this solution is explained some preliminary definitions are needed [28].

Definition 3.2.1: The Rank of an $m \times n$ matrix A, is the maximal number of linearly independent rows or columns. Rank(A) $\leq$ min{m,n}

Definition 3.2.2: An Orthogonal Matrix is a real square matrix A, such that $A^{-1}=A^T$, $(A^TA = I_n)$.

Definition 3.2.3: Let $L:V \rightarrow V$ be a linear transformation of an n-dimensional vector space into itself (a linear operation on V). The number λ is called an eigenvalue of L if there exists a *nonzero* vector **x** in V such that L(**x**) = λ **x**

Definition 3.2.4: Every nonzero vector **x** satisfying this equation is then called an eigenvector.

Definition 3.2.5: The Singular Values of a matrix A are the square roots of the eigenvalues of A^TA.

Definition 3.2.6: Singular Value Decomposition

Let A be an $m \times n$ real matrix. Then there exist orthogonal matrices U of size m x m and V of size n x n such that $A = USV^T$,

Where S is an $m \times n$ matrix with non-diagonal entries all zero and $s_{1,1}, s_{2,2}, \ldots, s_{p,p}$, where $p = \min\{m,n\}$.

The singular values, then, are the diagonal values of the matrix *S*.

Solving the MDGP using the Singular Valued Decomposition requires a full set of consistent (error free) distances initially placed into a square symmetric distance matrix. As mentioned, however, they must be placed into an induced distance matrix, which is done by first assuming that we can find coordinates; $x_i = (x_{i,1}, x_{i,2}, x_{i,3}),\ 1 \le i \le n$; and by allowing the last atom coordinates to be the origin, $x_n = (0, 0, 0)$. We can do this because any correct set of coordinates will be unique with respect to translation and rotation and by placing one atom at the origin we give ourselves a reference. We then seek coordinates $x_1, x_2, \ldots, x_n$ so that the distance between all pairs of points satisfies the given distances.

Mathematically…

$$\|x_i - x_j\| = d_{i,j} \quad 0 \le i, j \le n \tag{2}$$

Equivalently,

$$\|x_i\|^2 = d_{i,0}^2 \tag{3}$$

$$\|x_i - x_j\|^2 = d_{i,j}^2 \quad 1 \le i, j \le n-1 \tag{4}$$

By expanding the second set of constraints we have,

$$\|x_i\|^2 - 2x_i^T x_j + \|x_j\|^2 = d_{i,j}^2 \quad 1 \le i, j \le n-1 \tag{5}$$

$$d_{i,0}^2 - d_{i,j}^2 + d_{j,0}^2 = 2x_i^T x_j \quad 1 \le i, j \le n-1 \tag{6}$$

We then define the induced distance matrix

$D = [D_{i,j}]$ (7)

Where, $D_{i,j} = (d_{i,0}^2 - d_{i,j}^2 + d_{j,0}^2)/2 \quad 1 \le i, j \le n-1$

Let X be an (n-1) × 3 matrix where,

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & x_{1,3} \\ x_{2,1} & x_{2,2} & x_{2,3} \\ \vdots & \vdots & \vdots \\ x_{n-1,1} & x_{n-1,2} & x_{n-1,3} \end{bmatrix}$$

We then have,

$$D = XX^T \quad (8)$$

Here we may note that if the given distances are consistent then the equation will have a solution and must therefore be of rank ≤ 3. Knowing this we can find the singular valued decomposition of the n ×n induced distance matrix. This can be done in at most $O(kn^2)$ floating point operations [29] where k is the dimension of R^k. We then have,

$$D = U\Sigma U^T \quad (9)$$

Here, U is an $(n-1) \times 3$ matrix and Σ is a 3×3 diagonal matrix where the diagonal values are the largest singular values of D. Then,

$$D = U\Sigma U^T \quad (10)$$

$$D = U\Sigma^{\frac{1}{2}}\Sigma^{\frac{1}{2}}U^T = XX^T \quad (11)$$

We can then find a solution for X,

$$X = U\Sigma^{\frac{1}{2}} \quad (12)$$

The coordinates of the first (n-1) atoms are given by rows of the coordinate matrix X and the last atom is fixed at the origin. This method can also be generalized to solve the DGP when defined in k-space.

Theorem 3.2 [19]

Let $\{d_{i,j}: i,j = 1,\ldots,n\}$ be a set of distances in R^k, for some $k \leq n$. Then, the induced distance matrix D defined by equation (6) is of maximum rank k.

Proof: It follows from the fact that $D = XX^T$ for a coordinate matrix X in $R^{n-1} \times R^k$ and X is of maximum rank k because X is of size n-$1 \times k$ with $k \leq n$ thus the maximum number of linearly independent columns is at most k. ■

In k-dimensions the decomposition, $D = U\Sigma U^T$ is composed of U, which is an $(n\text{-}1) \times k$ matrix and Σ is reduced to a $k \times k$ diagonal matrix where the diagonal values are the singular values of the induced distance matrix D. For the DGP, Golub and van Loan showed that the singular value decomposition of a k-dimensional, $n \times n$ induced distance matrix would require $O(kn^2)$ floating point operations [29]. This would then have an upper bound of $O(n^3)$ again because $k \leq n$.

This solution has been employed by Crippen and Havel in there EMBED algorithm [17]. To be able to do this they must first estimate missing distances as we will see. Also, Sippl and Scheraga have developed an algorithmic solution to the DGP with a sparse set of distances by repeatedly finding the coordinates of complete subsets of distances [30]. With each repetition the number of coordinates increases and thus the

number of distances available increases because we can always compute missing distances from newly determined coordinates. The algorithm is completed when all atoms have been determined. These examples show that the simplified case of a complete set of exact distances has already led to multiple solutions for the more difficult case of a sparse/incomplete set of exact distances.

3.2.2 The MDGP with Sparse Exact Distances

In the case of the MDGP where an incomplete set of distances is all that is available, the problem is formulated in the same fashion as when a complete set of distances is given.

MDGP: Sparse Exact Distances

Let $a_1, a_2 \ldots a_n$ be the atoms of a protein and $d_{i,j}$, be the inter-atomic distances for complete set S of the pairs of atoms (a_i, a_j) $1 \le i, j \le n$. The problem then is to find coordinates in 3-dimensions $x_i = (x_{i,1}, x_{i,2}, x_{i,3})$ such that the distances between pairs of points x_i and x_j, given by the Euclidean norm (||.||), will satisfy the given distances.

$$\left\| x_i - x_j \right\| = d_{i,j} \quad (i, j) \subset S \tag{13}$$

As this sub-problem is the topic of this thesis, more will be said of it in the next chapter when discussing the geometric build-up solutions. Note, though, that this sub-problem is more closely related to the real-world case in that all pair-wise distances are not available. Recall that in either experimental method, X-ray or NMR, all atoms are not observable and thus only a subset of the possible inter-atomic distances can be recorded. Reducing the most practical case of the MDGP into these relatively simpler sub-problems

allows for a more thorough understanding of the problem as a whole. This also allows for the development of different solutions which can then be extended or adapted to handle the more practical case.

3.3.3 DGP with Distance Constraints or Inconsistencies

In practice, the experimentally collected distance data contains errors. Also, as stated in Chapter 2, NMR only provides lower and upper bounds on the inter-atomic distances. Therefore, the MDGP has been be modified to allow for error, specifically to account for the lower, $l_{i,j}$, and the upper, $u_{i,j}$, bounds. Formally the problem is formulated as follows.

MDGP: Distance Ranges

Let $a_1, a_2 \ldots a_n$ be the atoms of a protein and $l_{i,j}$ and $u_{i,j}$, be the lower and upper bounds on the inter-atomic distances for subset S of the pairs of atoms (a_i, a_j) $1 \le i, j \le n$. The problem then is to find coordinates in 3-dimensions $x_i = (x_{i,1}, x_{i,2}, x_{i,3})$ such that the distances between pairs of points x_i and x_j, given by the Euclidean norm (||.||), will satisfy the given distance constraints.

$$l_{i,j} \le \left\| x_i - x_j \right\| \le u_{i,j} \quad (i, j) \subseteq S \tag{14}$$

Similarly, we can formulate the problem to allow for some errors, or inconsistencies in the distance data as follows.

$$\left| \left\| x_i - x_j \right\| - d_{i,j} \right| \le \varepsilon_{i,j} \quad (i, j) \in S \tag{15}$$

A solution to this problem is known as an ε-approximate solution and the best of these is known as the ε-optimal solution [27][31].

Currently the major methods of solving the MDGP are the Singular Value Decomposition which requires a complete set of exact, consistent distances. In practice, however, the distance information has errors and is also incomplete. Two methods developed to account for this are the global optimization methods and the EMBED algorithm. Because this solution exploits the missing and inconsistent distances and utilizes both the SVD and the global optimization methods I offer the following brief description of the EMBED algorithm.

EMBED Algorithm

This algorithm takes on the practical case of the MDGP; therefore the input is typically a sparse case of distance ranges. The EMBED algorithm has three major stages, (1) bound smoothing, (2) distance metrication, and (3) global optimization [17][32]. The bound smoothing stage uses certain geometric properties like the triangle inequality to create distance ranges for the missing pairs of atoms. For example, if for three atoms, *i, j,* and *k*, two out of three sets out of distance ranges, $(l_{i,j}, u_{i,j}), (l_{i,k}, u_{i,k})$, are available, then using the triangle inequality the missing distance range $(l_{j,k}, u_{j,k})$ can be found. It follows from the fact that $d_{i,j} \le u_{i,j}$ and $d_{i,k} \le u_{i,k}$ that…

$$d_{j,k} \le d_{i,j} + d_{i,k} \le u_{i,j} + u_{i,k}$$

Therefore, $u_{j,k}$ can be replaced with the maximum value $u_{i,j} + u_{i,k}$. Similarly, $l_{j,k}$ can be found by using the inverse triangle inequality. Once this is done for all missing pairs of

atoms and a complete set of $n(n-1)/2$ distance ranges is generated for all n atoms in a protein, the distance metrication stage finds exact distances that fall within the constraints. These distances may be chosen as the midpoint of the range or may be calculated more appropriately by again incorporating the triangle inequality and shortest path trees. Once a complete set of exact, consistent distances is satisfactorily obtained the SVD method is used to calculate a set of coordinates. This set of coordinates is then used as a starting point for the final stage of the EMBED algorithm, global optimization. In fact the MDGP can be formulated as a global optimization problem using a least squares error function. To measure the distance between the coordinates and the given distance the relative error is defined as the following,

$$\varepsilon_R = \frac{\left\|x_i - x_j\right\|^2 - d_{i,j}^2}{d_{i,j}^2}, \quad (i,j) \in S \tag{16}$$

We then define an error function [27],

$$f(x_1,...,x_n) = \sum_{(i,j)\in S}\left[\frac{\left\|x_i - x_j\right\|^2 - d_{i,j}^2}{d_{i,j}^2}\right]^2 \tag{17}$$

Then if $X = (x_1,\ldots, x_n)$ is a solution the function will be equal to zero. This stage may be repeated many times with different sets of coordinates depending on the method used to choose the distances in the distance metrication stage. The minimization may be done using gradient methods or simulated annealing and thus a more accurate set of coordinates can be obtained. For a more detailed description of the EMBED algorithm see Crippen and Havel's book *Distance Geometry and Molecular Conformation* [17].

Chapter 4: Geometric Build-Up Algorithms (MDGP)

4.1 Introduction to the Geometric Build-Up Solution

Now that the importance of the MDGP in the area of protein determination has been defended and the problem itself explained, focus is shifted to the solution of the problem. As mentioned, the major solutions to the MDGP are the Singular Value Decomposition, global minimization, and the EMBED algorithm. Because these solutions are run computationally if the distances are inconsistent, not only will the SVD method fail but it will not be able to isolate the place where it fails. Also, these methods can be quite costly with regards to running time. Recall that the SVD method requires at most $O(n^3)$ floating point operations. The EMBED algorithm is also very costly. It's first two stages, bound smoothing and distance metrication can be very costly, in the order of $O(n^3 \sim n^4)$, and it also repeats the SVD step by determining different sets of exact distances satisfying the distance ranges. To improve on this running time Dong and Wu developed the so called geometric build-up (GBU) algorithm for a set of exact distances [1]. Since the method was first described and shown to be an improvement in the running time there has been a considerable amount of work done to improve and exploit other aspects of the solution as well. In this chapter we will review a set of related geometric build-up (GBU) solutions that have been developed, discussing the benefits and drawbacks of each solution as we explain the evolution to the novel method proposed in the project.

To understand the foundations of the method we will begin with a 2-dimensional example. Considering three points (x_1, x_2, x_3) in the plane and a set of given inter-atomic distances in a matrix form.

$$\begin{bmatrix} 0 & d_{1,2} & d_{1,3} \\ d_{1,2} & 0 & d_{2,3} \\ d_{1,3} & d_{2,3} & 0 \end{bmatrix}, \quad \text{where } d_{i,j} = \left\| x_i - x_j \right\|, \quad (i,j) \in \{1,2,3\}$$

With this information we can immediately find coordinates for three of the points as follows.

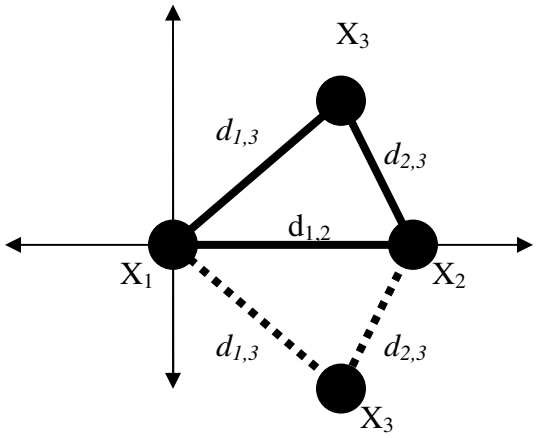


Fig. 1 Illustration for finding coordinates for 3 atoms in 2-D satisfying the distances

This is accomplished by placing the first point (x_1) at the origin and the second point (x_2) along an axis at the distance between the first and second point ($d_{1,2}$). The third point can then be placed in the plane in one of two positions that are actually reflections of one another. We can always choose the positive position for simplicity and this will not affect the coordinates' ability to fit the distance constraints. These three points are unique with respect to translation and rotation, and as long as they are not collinear these coordinates can be used to find the coordinates of a fourth point, again by using the given distance information.

To help explain the previous example the following definitions generalized to *k*-dimensions will be scaled back to both two and three-dimensions. The new terms will then be applied to the previous example and to the MDGP (3D) to show exactly how the coordinates are found.

Definition 4.1: A Metric Basis is a set of points B in a space S is a metric basis of S provided any point in S Can be uniquely determined by its distances to the points in B.

Definition 4.2: An Independent Set of *k*+1 Points is a set of *k*+1 points in R^k is called an independent set of points if it is not a set of points in R^{k-1}.

The connection, to the 2-dimensional example, is that we could find coordinates for the three points based only on their distance to one another, so they are a Metric Basis. As stated, if the points in the example are not collinear that set can be used to determine other points. Therefore, if we wish to determine the coordinates of another point the three points in the example must also be an independent set of points because they are a set of three points in R^2 that are not a set of three points in R^1.

Theorem 4.1: [3] A set of 4 independent points in R^3 form a metric basis for R^3.

Proof:

In 3-dimensional Euclidean space, let $x_i = (x_{i,1}, x_{i,2}, x_{i,3})^T$ be the coordinate vectors of 4 independent points i = 1, 2, 3, 4. Let $x_j = (x_{j,1}, x_{j,2}, x_{j,3})^T$ be the coordinate vector for any point j in R^3 with distances $d_{i,j}$ from points i = 1, 2, 3, 4 to point j. Then,

$$\left\| x_i - x_j \right\| = d_{i,j} \tag{18}$$

Equivalently,

$$\left\| x_i \right\|^2 - 2x_i^T x_j + \left\| x_j \right\|^2 = d_{i,j}^2 \tag{19}$$

By solving each equation for x_j, we obtain the system of equations,

$$Ax_j = b$$

Where, $A = -2\begin{bmatrix} (x_2 - x_1)^T \\ (x_3 - x_2)^T \\ (x_4 - x_3)^T \end{bmatrix}$, and $\quad b = \begin{bmatrix} (d_{2,j}^2 - d_{1,j}^2) - (\left\| x_2 \right\|^2 - \left\| x_1 \right\|^2) \\ (d_{3,j}^2 - d_{2,j}^2) - (\left\| x_3 \right\|^2 - \left\| x_2 \right\|^2) \\ (d_{k+1,j}^2 - d_{k,j}^2) - (\left\| x_{k+1} \right\|^2 - \left\| x_k \right\|^2) \end{bmatrix}$

Then, because the 4 points are an independent set they are not a set of points in R^2 the matrix A must be non-singular and is thus invertible. Thus the coordinate vector for x_j is uniquely determined by

$$x_j = A^{-1} \cdot b \qquad \blacksquare$$

Based on the above definitions and theorem we can then realize that if we have four atoms in 3-dimensions that are not in the same plane (R^2), we can immediately find the coordinates for those atoms in 3-dimensional Euclidean space. We will now show how the coordinates of the four "base" atoms (the four atoms that make up the metric basis) can be used to determine additional atoms. The build-up process begins by finding the coordinates of the four base atoms, $x_i = (u_i, v_i, w_i)$, $i = \{1,2,3,4\}$, by placing the first

atom at the origin, $x_1 = (0,0,0)$, the second atom along the first axis so that the distance between the two is satisfied, $x_2 = (d_{1,2},0,0)$. To determine the coordinates of the third atom we must solve the following equations by using the coordinates of the previous two.

Here we assume that x_1, x_2, and x_3 are in the x-y plane

$$\begin{aligned} &u_3^2 + v_3^2 = d_{1,3}^2 \\ &(u_3 - u_2)^2 + v_3^2 = d_{2,3}^2 \\ &w_3 = 0 \end{aligned} \tag{20}$$

After solving, we have,

$$\begin{aligned} u_3 &= (d_{1,3}^2 - d_{2,3}^2)/(2u_2) + u_2/2 \\ v_3 &= \pm\sqrt{(d_{1,3}^2 - u_3^2)} \\ w_3 &= 0 \end{aligned} \tag{21}$$

We choose v_3 to be positive, as it will not change the accuracy of future determinations. The fourth atoms coordinate vector is similarly obtained by using the previous coordinates to solve the following.

$$\begin{aligned} &u_4^2 + v_4^2 + w_4^2 = d_{1,4}^2 \\ &(u_4 - u_2)^2 + v_4^2 + w_4^2 + d_{2,4}^2 \\ &(u_4 - u_3)^2 + (v_4 - v_3) + w_4^2 + d_{3,4}^2 \end{aligned} \tag{22}$$

Solving for $x_4 = (u_4, v_4, w_4)$ we get,

$$\begin{aligned} u_4 &= (d_{1,4}^2 - d_{2,4}^2)/(2u_2) + u_2/2 \\ v_4 &= (d_{2,4}^2 - d_{3,4}^2 - (u_4 - u_2)^2 + (u_4 - u_3)^2)/(2v_3) + v_3/2 \\ w_4 &= \pm\sqrt{(d_{1,4}^2 - u_4^2 - v_4^2)} \end{aligned} \tag{23}$$

Again, we allow the third coordinate of the fourth atom to be positive as it does not change the resulting structure, although an exact mirror image structure may be obtained by making all third coordinates negative.

We now have the positions of $x_i = (u_i, v_i, w_i)$, $i = \{1,2,3,4\}$, which we will refer to as the base atoms coordinate vectors. If we then have all distances from the base atoms to another atom (x_j) in the protein we can set up a system of equations as follows.

$$\begin{aligned} \|x_j - x_1\| &= d_{j,1} \\ \|x_j - x_2\| &= d_{j,2} \\ \|x_j - x_3\| &= d_{j,3} \\ \|x_j - x_4\| &= d_{j,4} \end{aligned} \tag{24}$$

Squaring both sides we obtain,

$$\begin{aligned} \|x_j\|^2 - 2x_j^T x_1 + \|x_1\|^2 &= d_{j,1}^2 \\ \|x_j\|^2 - 2x_j^T x_2 + \|x_2\|^2 &= d_{j,2}^2 \\ \|x_j\|^2 - 2x_j^T x_3 + \|x_3\|^2 &= d_{j,3}^2 \\ \|x_j\|^2 - 2x_j^T x_4 + \|x_4\|^2 &= d_{j,4}^2 \end{aligned} \tag{25}$$

We can then expand the middle terms on the left to get,

$$\begin{aligned} \|x_j\|^2 - 2u_j u_1 - 2v_j v_1 - 2w_j w_1 + \|x_1\|^2 &= d_{j,1}^2 \\ \|x_j\|^2 - 2u_j u_2 - 2v_j v_2 - 2w_j w_2 + \|x_2\|^2 &= d_{j,2}^2 \\ \|x_j\|^2 - 2u_j u_3 - 2v_j v_3 - 2w_j w_3 + \|x_3\|^2 &= d_{j,3}^2 \\ \|x_j\|^2 - 2u_j u_4 - 2v_j v_4 - 2w_j w_4 + \|x_4\|^2 &= d_{j,4}^2 \end{aligned} \tag{26}$$

To solve the equation we can subtract the first equation from all other three.

$$\begin{aligned} 2u_i(u_1-u_2)+2v_i(v_1-v_2)+2w_i(w_1-w_2) &= (\|x_1\|^2-\|x_2\|^2)-(d_{j,1}^2-d_{j,2}^2) \\ 2u_i(u_1-u_3)+2v_i(v_1-v_3)+2w_i(w_1-w_3) &= (\|x_1\|^2-\|x_3\|^2)-(d_{j,1}^2-d_{j,3}^2) \\ 2u_i(u_1-u_4)+2v_i(v_1-v_4)+2w_i(w_1-w_4) &= (\|x_1\|^2-\|x_4\|^2)-(d_{j,1}^2-d_{j,4}^2) \end{aligned} \quad (27)$$

This system of equations can then be written in the form

$$Ax_j = b \quad (28)$$

where, $A = 2\begin{bmatrix} u_1-u_2 & v_1-v_2 & w_1-w_2 \\ u_1-u_3 & v_1-v_3 & w_1-w_3 \\ u_1-u_4 & v_1-v_4 & w_1-w_4 \end{bmatrix}$, and $b = \begin{bmatrix} (\|x_1\|^2-\|x_2\|^2)-(d_{j,1}^2-d_{j,2}^2) \\ (\|x_1\|^2-\|x_3\|^2)-(d_{j,1}^2-d_{j,3}^2) \\ (\|x_1\|^2-\|x_4\|^2)-(d_{j,1}^2-d_{j,4}^2) \end{bmatrix}$

The solution can be found by multiplying both sides of the equation by the inverse of A.

$$x_j = A^{-1}b$$

Here we see that we require the inverse of a square matrix. If the four base atoms are in the same plane then the matrix A will be singular and the inverse of A will not be available. In practice proteins will have a large number of atoms from which to choose the base set and therefore, singularity is usually an avoidable issue. We will now begin our discussion of the evolution of the geometric build-up solution.

4.2 All Exact Distances

Dong and Wu first described the Geometric Build-Up (GBU) method as it applies to the case of a complete set of exact inter-atomic distances [1], and it should be clear that if all distances are available and we choose a set of base atoms, we will certainly have the

distance from each base atom to all remaining atoms in the protein. Therefore, an algorithm has been developed that looks at each undetermined atom and uses the four distances to the base atoms to determine unique coordinates for the each of the remaining atoms. Because all distances needed are available at each step in the algorithm, it will only require $O(n)$ floating point operations for a protein with n atoms. This is a great improvement on the SVD step used be Crippen and Havel [17][33] which requires $O(n^3)$. As great as this is, it is not the only improvement. Dong and Wu also explain that the geometric build-up allows for close inspection of each step whereas the SVD handles more information behind the scenes. This means that the GBU can detect inconsistencies in the distance data because at each step a system of equations must be solved which, if the data is inconsistent, will not have a solution and immediately the trouble causing distance(s) can be found among the four presently being used. SVD on the other hand will fail when given inconsistent data but will not give any indication where the failure took place. Another great thing about this algorithm is that it may, in the future, be adapted to directly apply to the distance intervals that are obtained from NMR experiments. This could, potentially, eliminate the distance metrication stage of the EMBED algorithm, where coordinates are found to satisfy the constraints, which are then minimized in the third and last stage. Moreover, the GBU solution actually employs only four distances per atom. This gives a clue that we may not even have to estimate all of the missing distance intervals, the costly bound smoothing stage of the EMBED algorithm. We now show that the GBU can handle some very sparse distance data as we will see.

4.3 Sparse Exact Distances

In addition to the benefits above the GBU does only use a small portion of the distances available. In fact, the determination of each atom uses only four of the (n-1) distances available for it. To be exact the four base atoms require only six distances and each atom thereafter requires four distances each, so in total only (4n-10) distances are used from a total of (n^2-n)/2. This is probably the best evidence that this algorithm can also work for a sparse set of distances. In fact, Wu, Wu, and Yuan very nicely gave the following necessary and sufficient conditions for the GBU generalized to k-dimensions.

Theorem 4.2: [31] A necessary condition for the unique determination of the coordinates of a group of points $x_1,\ldots,x_n$ in R^k with a given set of distances among the points is that each point must have at least k+1 distances from other k+1 points, assuming that this point is not in R^{k-1} with any k fo the $k+1$ points.

Proof:

It follows immediately from the fact that in R^k, a point can be defined uniquely only if it has k+1 distances from k+1 independent points, assuming it is not in R^{k-1} with any k of the k+1 points. If is has only k distances from k points, the point will have two reflective positions. ■

Theorem 4.3: [31] A sufficient condition for the unique determination of the coordinates of a group of points $x_1,\ldots,x_n$ in R^k with a given set of distances among the points is that in every step of the geometric build-up algorithm, there is an undetermined point with k+1 distances from k+1 independent and determined points.

Proof:

The geometric build-up algorithm gives a constructive proof for the theorem, because if the condition holds in every step of the algorithm, the algorithm will be able to determine the coordinates of all the points uniquely. ■

In 3-dimensions this just means that if each atom does not have at least four distances to four other atoms the GBU will fail to determine a complete set of coordinates every time. These four distances are necessary but they are not enough to guarantee a complete structure. This may be due to the planarity of the four base atoms or because some of the distances may be to undetermined atoms. For example, if there are two atoms remaining and they each have three distances to already determined atoms but the fourth distance is between the pair of undetermined atoms, then the algorithm will fail to find a satisfactory solution. To be sufficient we must, in every step of the build-up, have an undetermined atom with at least four distances to four determined atoms. Dong and Wu were the first to use the GBU in an algorithm for sparse sets which required $O(n^4)$ floating point operations. This is not a lower bound, as we will see, but their initial algorithm is the following.

A GBU Algorithm for MDGP with Sparse sets of Distances

1. F = {*four initial atoms*}; fixed atoms
2. $\underline{U}$ = {n-4 *atoms*}; unfixed atoms
3. **while** $U \neq \phi$ **do**

A. **for** $a \in U$ **do**

i. find b_1, b_2, b_3, b_4 in F; distances to a available

ii. fix a with b_1, b_2 ,b_3 ,b_4:

iii. $F = F \cup \{a\}$; U $= U - \{a\}$: move a from U to F

B. **end**

C. **if** no a in U is fixed, stop; structure partially determined

4. **end**

5. structure completely determined

Fig. 2 The outline of the GBU algorithm for solving the MDGP with sparse distances. [2]

The for and while loops in the above algorithm each require n steps. It also requires at most n steps to find the four base atoms, because there are at most n determined atoms to choose from, and at most another n steps to check that they have all the distance to the undetermined atom. This total running time of $O(n^4)$ can be reduced with the help of different data structures and programming techniques as will be shown later. The computational complexities of this problem become varied and many and despite the many benefits they have brought about, computers are not without their limitations, namely memory. Numerical values stored in computers cannot have an infinite number of digits, therefore they must be truncated which introduces the problem of round off error.

It should be noted that to develop the algorithms it is important to have known structures and create test cases from them to work with. Creating the test cases is accomplished by using known 3D structures from the Protein Data Bank (PDB) online,

which stores information critical to the proteins structure and function. This information includes the number and type of atoms, the method used to determine the necessary data, the authors of the structure, etc., and also a 3-dimensional image that can be viewed using different modeling software. To produce the image the PDB stores coordinates for each of the atoms in the molecule. These coordinates can be downloaded as part of a PDB file and distances can then be computed. Specifically, to create a sparse set of data and to more accurately reflect actual NMR results a cutoff distance is established at say five angstroms (5Å). NMR is only able to determine distances for atoms that are close enough together that their spin state is affected. Then all distances greater than or equal to the cutoff can be removed from the distance matrix, the lower the cut-off the fewer available distances. Furthermore, we can calculate the error in the algorithms by comparing our new built-up coordinates to the original PDB coordinates using the Root Mean Square Deviation (RMSD). This is necessary in the case of sparse data, due to the propagation of the round off errors. This occurs because earlier determined atoms are used to determine later atoms. This is illustrated in Table 4.1 that shows my test results for a sparse algorithm programmed in Matlab, a scientific programming language in which all objects are stored as matrices.

4.4 Root Mean Square Deviation (RMSD) Error Calculation

Here we briefly describe the RMSD calculation of the error. If we have two sets of coordinates, X and Y, each in the form of an $n \times 3$ matrix. We first define the Root Mean Square Deviation (RMSD) as.

$$\text{RMSD}(X,Y) = \min_{Q} \left\| X - YQ \right\|_F / \sqrt{n} \tag{29}$$

Where, $\|.\|_F$ is the Frobenius norm

Definition 4.3 [29]: The Frobenius norm of a matrix *D*, denoted $\|D\|_F$, is a matrix norm similar to the Euclidean norm that is defined to be the square root of the sum of the absolute squares of the elements of *D*, $d_{i,j}$. That is,

$$\|D\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}\left|d_{i,j}\right|^2}$$

Q, then, is the rotation matrix such that $QQ^{\mathrm{T}} = \mathrm{I}$. To solve this optimization problem we will use a familiar technique, the singular value decomposition. The method follows.

Compute the geometric centers of the structures.

$$xc = \frac{1}{n}\sum_{i=1}^{n} X(i,:) = (xc_1, xc_2, xc_3) \qquad yc = \frac{1}{n}\sum_{i=1}^{n} Y(i,:) = (yc_1, yc_2, yc_3)$$

We then align the geometric centers by translating all sets of coordinates in both structures.

$$\begin{aligned} X(:,1) &= X(:,1) - xc_1 & Y(:,1) &= Y(:,1) - yc_1 \\ X(:,2) &= X(:,2) - xc_2 & Y(:,2) &= Y(:,2) - yc_2 \\ X(:,3) &= X(:,3) - xc_3 & Y(:,3) &= Y(:,3) - yc_3 \end{aligned}$$

Then to calculate Q, let $C = X^T Y$. We then use the SVD to find $C = U\Sigma V^T$, and thus $Q = U\Sigma^{\frac{1}{2}}$ is the solution to the minimization problem.

In my sparse case algorithm, different methods have been attempted to choose a base set of atoms so that they will have distances to more than one undetermined atom and can therefore be use to determine multiple atoms before a new set must be found. The chosen method first creates an adjacency matrix of ones and zeros from a sparse distance matrix. If a distance, $d_{i,j}$, is available the corresponding cell of the adjacency matrix, $a_{i,j}$, contains a one and if the distance is not available it will contain a zero. This method then goes on to sum all sets of four consecutive rows of the adjacency matrix to create a new matrix of size (n-3 × n). The numbers of fours in each row of the new matrix are counted to find the consecutive set of four that has the most atoms with distances to all four. The maximum of this array is used to indicate which first four atoms to use as a base set to be determined using the GBU methods already described. Once a base set is chosen all atoms with deistances to it can be determined. These steps are all taken prior to the while loop and they initially set up the lists of determined and undetermined atoms. This method can, however, be modified for use within the for and while loops.

In each iteration of the algorithm a base set of atoms must be found from the determined atoms. Therefore, a base set can be found by creating, from the original adjacency matrix, a modified adjacency matrix. Note that each atom is numbered according to their order in the matrices (all consistent) and the lists of determined and undetermined atoms are referenced by their atom numbers. For this reason the modified adjacency matrix can be created by using list of determined atoms as the row numbers and the list of undetermined atoms as the column numbers. Then, in summing the four consecutive rows a determined base set can be found with a maximal number of undetermined atoms having all necessary distances to the base set.

This method, however, can cause varied results in the error depending on the arrangement of the atoms in the protein and the order of the atoms in the matrix. It may fail because the four required distances in a step may not be found to be consecutive in the given ordering. Table 4.1 illustrates both the propagation of errors as well as the uncertainty of the algorithms stability in controlling the errors. To understand this, consider a protein that is dense in the sense that the atoms are tightly packed together. Then, with each new choice of a base set there will likely be many distances available to undetermined atoms because more distances will be less than the cut-off. This can result in less running time mainly because fewer base sets will need to be found.

Protein Name (PDB)	Size (#Atoms)	RMSD (Error)
1FW5	332	1.2574 Å
1CEU	854	0.8286 Å
1CTO	1739	550.996 Å

Table 4.1: sparse cutoff at 5 Å. The error increases dramatically for the larger of the proteins 1CTO as compared to the smallest 1FW5. The other protein, 1CEU, illustrates that using only consecutive rows for finding base sets does not provide a constant rate of error increase due to the larger atomic size but the smaller error.

How can these errors be controlled and minimized? To answer this question Wu and Wu developed what is sure to become a standard piece of the GBU solution. They dubbed his addition to the GBU the Updating routine as part of their so called Updated Geometric build-up solution.

4.5 The Updating Routing For the GBU with Sparse Exact Distances

Error propagation posed a serious obstacle for the evolution of the GBU. To understand how this happens imagine tuning a piano. If the first string is tuned using a tuning fork or pitch pipe and each string thereafter is tuned using the string previous to it how accurate or inaccurate would the 88^{th} string be? If, however, a tuning fork or pitch pipe were used throughout the process each string will be closer to its actual frequency and thus minimize the overall error. Relative to the MDGP the tuning fork can represent the given distance data and similar to tuning the piano the more often we use original, accurate data the better able we are to keep track of and minimize the errors of each atom.

It was Di Wu and Z. Wu who first developed the Updated Geometric Build-Up Algorithm [3]. They described an algorithm that chooses a base set of determined atoms that also has a complete subset of inter-atomic distances in the original distance matrix. Recall from Theorem 4.1 that if all inter-atomic distances between four non-coplanar atoms are known, coordinates for the four base atoms can immediately be found for them in an independent coordinate system. Two sets of coordinates are then obtained; one set from the general build-up process and another independent and updated set. These two structures can be visualized as two tetrahedral in 3-space. The next step of the updating

routine is to calculate the geometric centers of the two structures and then place the updated coordinates into the GBU structure by aligning their geometric centers. The atoms can then be adjusted by rotating about the common center of mass so that error is minimized, as measured using the Root Mean Square Deviation (RMSD).

Mathematically, let X be the coordinates obtained from the general GBU and Y be the new coordinates obtained by fixing the chosen four base atoms in an independent system using the necessary six distances provided in the original distance data.

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & x_{1,3} \\ x_{2,1} & x_{2,2} & x_{2,3} \\ x_{3,1} & x_{3,2} & x_{3,3} \\ x_{4,1} & x_{4,2} & x_{4,3} \end{bmatrix} \qquad Y = \begin{bmatrix} y_{1,1} & y_{1,2} & y_{1,3} \\ y_{2,1} & y_{2,2} & y_{2,3} \\ y_{3,1} & y_{3,2} & y_{3,3} \\ y_{4,1} & y_{4,2} & y_{4,3} \end{bmatrix}$$

Recall the aforementioned Root Mean Square Deviation (RMSD).

$$\text{RMSD}(X,Y) = \min_{Q} \left\| X - YQ \right\|_F / \sqrt{n} \tag{30}$$

Where, $\left\| . \right\|_F$ is the Frobenius norm and Y is the matrix of the new translated coordinates. Q is the rotation matrix such that $QQ^{\mathrm{T}} = \mathrm{I}$.

We compute the geometric centers,

$$xc = \frac{1}{4}\sum_{i=1}^{4} X(i,:) = (xc_1, xc_2, xc_3) \qquad yc = \frac{1}{4}\sum_{i=1}^{4} Y(i,:) = (yc_1, yc_2, yc_3)$$

A translation vector can be obtained by simply subtracting *xc* from *yc*, then translating *Y* by subtracting the translation vector.

That is, find a translation vector.

$$yc - xc = (tv_1, tv_2, tv_3)$$

Translate the independent set of coordinates *Y*,

$$Y = \begin{bmatrix} (y_{1,1} - tv_1) & (y_{1,2} - tv_2) & (y_{1,3} - tv_3) \\ (y_{2,1} - tv_1) & (y_{2,2} - tv_2) & (y_{2,3} - tv_3) \\ (y_{3,1} - tv_1) & (y_{3,2} - tv_2) & (y_{3,3} - tv_3) \\ (y_{4,1} - tv_1) & (y_{4,2} - tv_2) & (y_{4,3} - tv_3) \end{bmatrix}$$

We then set up the same optimization problem as in the RMSD error calculations. That is,

$$\text{RMSD}(X,Y) = \min_Q \left\| X - YQ \right\|_F / \sqrt{n}$$

Where, $\left\| . \right\|_F$ is the Frobenius norm and *Y* is the matrix of the new translated coordinates. *Q* is the rotation matrix such that $QQ^T = I$. We can, again, use the SVD to solve for *Q* by letting $C = X^T Y$, then we find $C = U\Sigma V^T$ and then we must have $Q = U\Sigma^{\frac{1}{2}}$. Just as in calculating the RMSD for error we are simply rotating the structure about a point in 3-space and minimizing the error between the two.

The Updated GBU Algorithm for MDGP with Sparse Distances

1. F = {*four initial atoms*}; determined atoms

2. U = {n-4 *atoms*}; undetermined atoms

3. **while** $U \neq \phi$ **do**

    A. **for** $a \in U$ **do**

        i. find b_1, b_2, b_3, b_4 in F; distances to a available

        ii. determine a with b_1, b_2, b_3, b_4

        iii. **if** all distances between atoms b_1, b_2, b_3, b_4 are known

            (a) update the coordinates of b_1, b_2, b_3, b_4 and a

            (b) replace atoms in original structure with updated coordinates.

        iv. **end**

        v. **if** no a in U is determined, stop; structure partially determined

    B. **end**

4. **end**

5. structure completely determined

Fig. 3 The outline of the Updating GBU algorithm for solving the MDGP with sparse exact distances [3].

This is very beneficial because now we have a set of coordinates that are definitely more accurate not only relative to one another but also to the overall structure. Therefore, by repeating this updating routine and continually replacing the built-up, error induced coordinates with the updated coordinates, existing error can be corrected and the transmission of errors to atoms yet to be determined can be prevented. Numeric test results presented in section 4.6 will then show how this has affected the solution of the

GBU. Suffice it to say that this innovative development to the GBU has proven to be an excellent solution to the problem of error propagation.

Another stage in the evolution of the GBU returns to the issue of sparse data. Some of the earliest test cases derived used cut-off distances of 10Å and 8Å, but as mentioned the smaller the cut-off the fewer available distances and, thus, the more sparse the data. GBU algorithms developed more recently are able to handle cases with very few distances by using an idea: for an undetermined atom, if only three distances to three determined atoms are known then two reflective sets of coordinates for the undetermined atom can be determined.

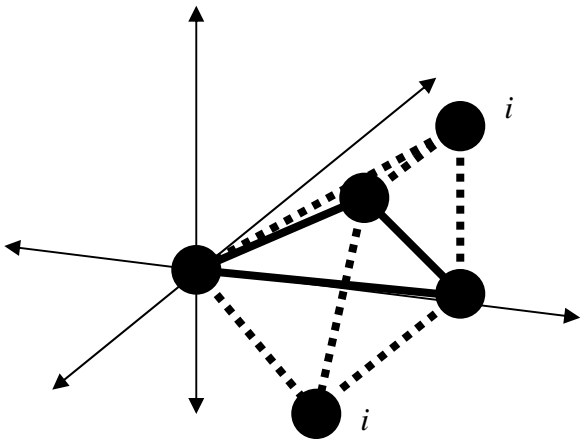


Fig. 4 Illustration of the rigid determination.
Atom *i* has two possible positions

This is the idea behind the Rigid Branch and Prune Algorithm which was also described by Wu, Wu, and Yuan [31] [34]. This is also a method used in the algorithm of this thesis. However, here it is employed, not as a method to handle cases with fewer distances, but rather as a way to speed up the process of finding a base set of atoms. It is

certainly faster to find three atoms with all three inter-atomic distances than it is to find four atoms with all six inter-atomic distances.

4.6 Revised Updated GBU Solution

The previous chapters and sections have been a survey of the works of many biologists, chemists, physicists, and mathematicians, and it has all led to this the latest algorithm developed in the progression of the GBU method. This algorithm, known as the Revised Updated Geometric Build-up (RUGB) algorithm, has been developed by combining all of the GBU variations discussed previously. The goals of this algorithm are improve accuracy, relative to the general GBU solution, by employing the Updating routine as well as improve the running time of the GBU solution. The updating routine has already been shown to improve the accuracy of the GBU solution [3] while it does not significantly affect the running time.

In programming the sparse case GBU the computational complications in choosing the base set of atoms quickly become apparent. The largest complication and the main hindrance to the running time is the search for a base set of atoms. Finding four base atoms with all distances to the undetermined atom is, in itself, difficult and time consuming. This says nothing about trying to find the base set with the most number of undetermined atoms with all distances to the base atoms or finding a base set with a complete set of inter-atomic distances for use in the updating routine. The answers to these difficulties are in the programming. Utilizing different data structures to exploit the properties of the overall structure as well as inputting the distance data and processing it in such a way that all of the pieces are readily accessible.

Now is a good time to introduce some graph theory terminology that will be used herein in reference to the MDGP. In graph theory two vertices are adjacent if there is an edge between them. Here adjacent is also used to mean the distance is available in the original distance information which corresponds to the GEP. Also, recall that for a base set of atoms to be used in the updating routine all distances between the base set of atoms must be available, it must be a complete subset. In graph theory a set of vertices with a complete set of edges among them is called a clique, thus for a unique determination def. 4.1 is referring to a k-clique. For the MDGP we require a 4-clique for the unique determination and a 3-clique or triangle for the rigid determination.

In searching for ways to best find and choose a 4-clique and looking into general k-clique detection methods and algorithms it was discovered that this too is a very complicated problem. In fact there are really no fast algorithms for finding k-cliques for $k > 3$, but there are some algorithms for finding triangles (3-cliques). These include both counting and listing algorithms and node-iterators and edge-iterators. The running times range according to the method, but it seemed for the purposes of the MDGP the best suited algorithm was a node-iterating triangle listing algorithm [35]. It is actually a brute force algorithm listing all of the available triangles that has a running time of $O(d_{\max}^2 n)$. Here d_{max} is the maximum number of distances available to any atom in the protein and this consideration of d_{max} is the idea behind the data structure improvement. Rather than searching all n atoms for adjacency, if the distance matrix is first analyzed an array of adjacency lists can be created. By referencing these lists we can quickly find a triangle so that each of its vertices (the atoms) are adjacent to the undetermined atom at hand. The three base atoms, because they form a 3-clique, can then be updated using the routine

described by Wu and Wu. At this point the undetermined atom can be determined rigidly in the independent coordinate system and because it will have two possible positions all five sets of coordinates can be placed into the structure by rotating and translating. A fourth determined atom adjacent to the rigidly determined atom can then be found by referencing the adjacency list once again. Using that distance data one of the two possible positions can then be eliminated if it does not satisfy the original data. Fig. 4.7 shows the pseudocode for this novel algorithm.

The Revised/Rigid GBU Algorithm

1. Find four base atoms that are not in the same plane
2. Determine the base atoms uniquely
3. While not all atoms are determined

   For each undetermined atom u_i

   Find a triangle with determined atoms having all distances to u_i

   If such a triangle exists

   Find a fourth atom with distance to u_i for use as a cut-off distance

   End

   If both a triangle and a cut-off atom are available

   Determine u_i and update using the described Updating Routine

   End
4. If no atom can be determined uniquely, stop.
5. All atoms are determined uniquely

Fig. 5 The outline of the Revised Updating GBU algorithm for solving the MDGP with sparse exact distances.

The running time for the RUGB is $O(d_{\max}^3 n^2)$ due to the use of only two major loops and the new data structures used in finding a base set. More specifically, the for and while loops each require at most n steps because there are a total of only n atoms, thus there can be at most n undetermined atoms. This is the reason for the inclusion of n^2 in the bound on the running time. Inside the for loop a triangle base set (b_1, b_2, b_3) must be found for an undetermined atom i. This is accomplished by looking, first, at the list of atoms adjacent to atom i, adjlist(i), to find b_1. The adjacency list for b_1, adjlist(b_1), is then used to find b_2 and subsequently adjlist(b_2) is used to find b_3. Each adjacency list is bounded by d_{max}, therefore these steps, within the for loop, are bounded by $d_{\max}^3$. Also, finding a fourth cut-off atom will require at most d_{max}. The newly employed data structure, an array of adjacency lists, speeds up the process of finding a base set of atoms but is also aided by the uses of a separate list of n ones and zeros, where the one indicates determined and zero indicates the atom is undetermined. This provides a way to verify that status with a single check. Also, the adjacency of all atoms in the triangle can be verified, each with a single check, while the adjacency lists are scanned.

As evidence of the running time Figure 6 plots the number of atoms squared, n^2, against the running time, in seconds, for 27 proteins testing the RUGB. It shows a very linear relationship and in fact the correlation coefficient (r = .9257) is quite high. The test results are illustrated further in Table 4.2 which lists the proteins' names as they appear in the Protein Data Bank along with the number of atoms. The running time, and the RMSD error resulting from the RUGB tests are also included in Table 4.2 as well as the minimum, d_{min}, and the maximum, d_{max}, degrees at a 5Å cutoff.

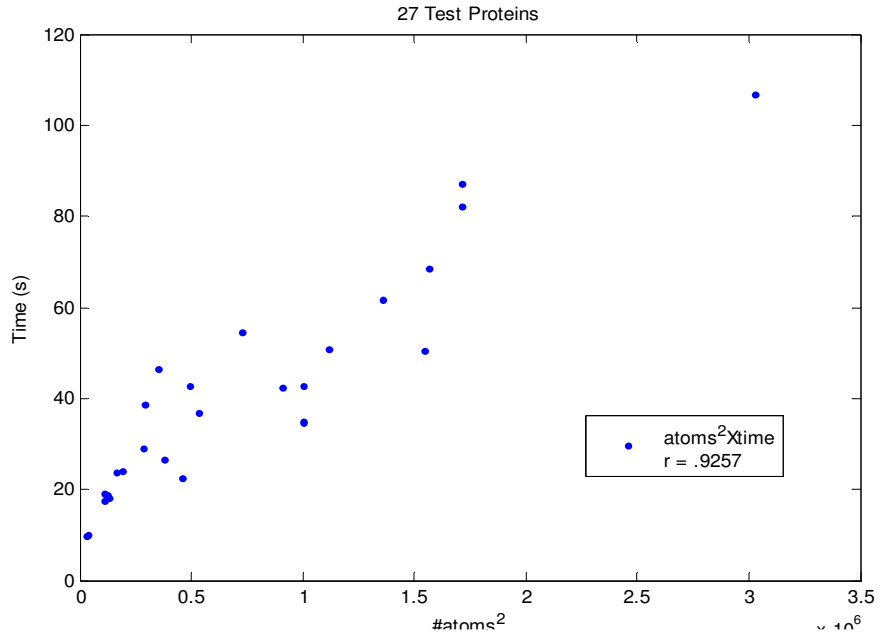


Fig. 6 Illustration of $O(n^2 d^3_{max})$ running time for the Revised Updated GBU algorithm

To be sure that this was the best fit for the running time the correlation coefficients for the following cases have been verified: corrcoef(n^3, time) = .8964 and corrcoef(n^4, time) = .844 and corrcoef(n, time) = .9147. Thus, the best correlation with a correlation coefficient of .9257 is that of n^2.

PDB Name	d_{min}	d_{max}	# atoms	Time (s)	RMSD error
1AIK	5	49	729	36.8232	8.59E-11
1BOM	9	69	700	42.8199	1.69E-06
1BWI	4	39	1001	42.7253	1.65E-07
1CEU	7	65	854	54.674	1.52E-10
1HAA	7	69	1310	87.1441	8.66E-08

1HLL	11	94	540	38.5727	8.42E-10
1HSM	10	73	1251	68.5916	1.21E-08
1IMQ	9	77	1308	82.0806	2.05E-04
1KVX	4	38	954	42.3962	2.60E-06
1LIH	4	41	1243	50.5037	4.88E-07
1R7C	9	64	532	28.9494	2.24E-09
1ULR	4	36	677	22.4415	3.30E-08
1VII	8	77	596	46.4245	3.33E-11
1VMP	10	74	1166	61.6084	7.98E-08
2DX2	8	58	174	9.7131	1.15E-11
2EZH	8	66	1058	50.8981	9.02E-09
6LYT	4	39	1001	34.8481	1.79E-08
8LYZ	4	39	1000	34.5145	2.57E-08
1CTO	8	74	1739	106.6881	9.38E-06
1SOL	9	60	353	18.8787	1.88E-12
1JAV	9	71	360	18.089	3.21E-06
1IDV	9	72	189	10.208	8.00E-14
1AMB	8	65	438	24.1381	6.61E-10
1B5N	10	67	332	17.4421	9.84E-08
1FW5	9	66	332	19.0297	5.25E-08
1HIP	5	37	617	26.5239	4.02E-08
1meq	10	72	405	23.859	5.02E-11

Table 4.2
27 proteins testing the running time and error propagation for the new Revised Updated GBU algorithm

To test the effectiveness of the RUGB algorithm in regards to error control, it was tested against a sparse algorithm with the updating routine (SparseN6), the sparse algorithm without the updating routine (SparseAlgorithm), as well as testing the RUGB without the updating routine. Table 4.3 list the running times and the propagated error build up for each of the 10 atoms and Figure 7 plots the error for the 10 proteins comparing the RUGB with and without the Updating Routine.

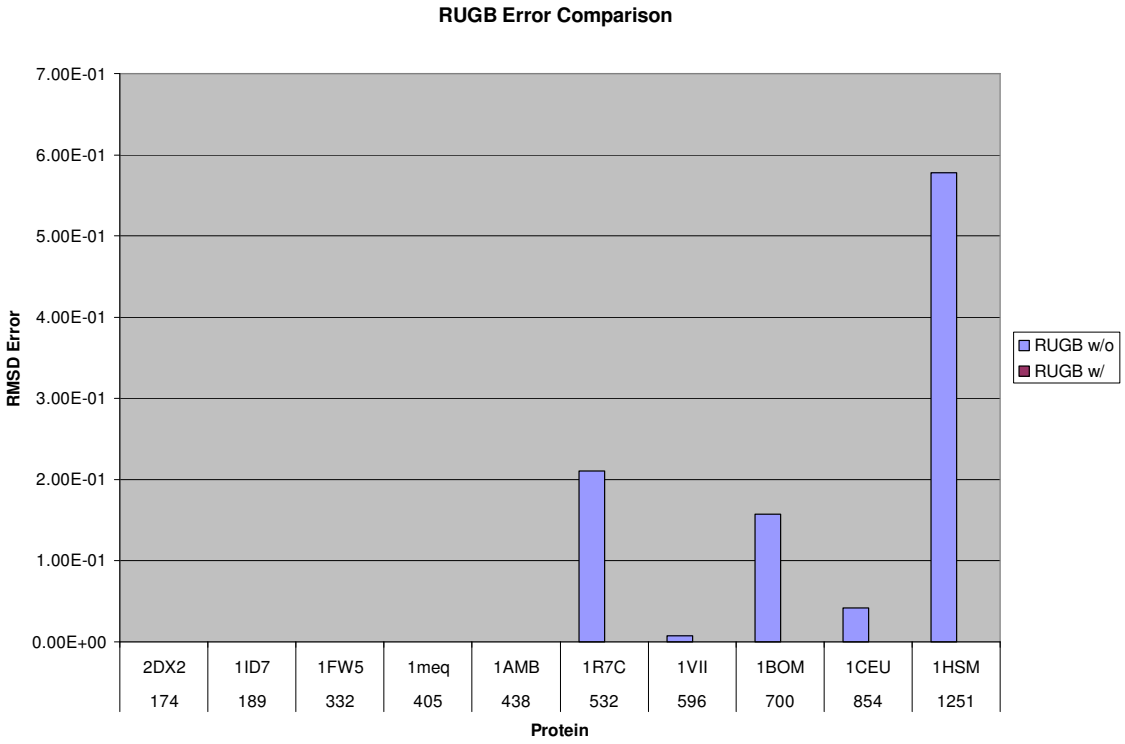


Fig. 7 Illustration of the effectiveness of the Updating Routine on the new Revised Updated GBU algorithm.

It is clear that as the number of atoms gets large, the error without the Updating routine quickly gets out of control. Also the largest of the 10 proteins was not determined uniquely when the updating routine was turned off, however, when the updating routine was turned on, not only was it able to determine the entire protein but it did so while keeping the error extremely close to zero. Figure 8 compares the two updating algorithms, the SparseN6 and the new RUGB. This shows that the total error for small proteins remains small for both algorithms, data for the Figure 8 is shown in Table 4.2.

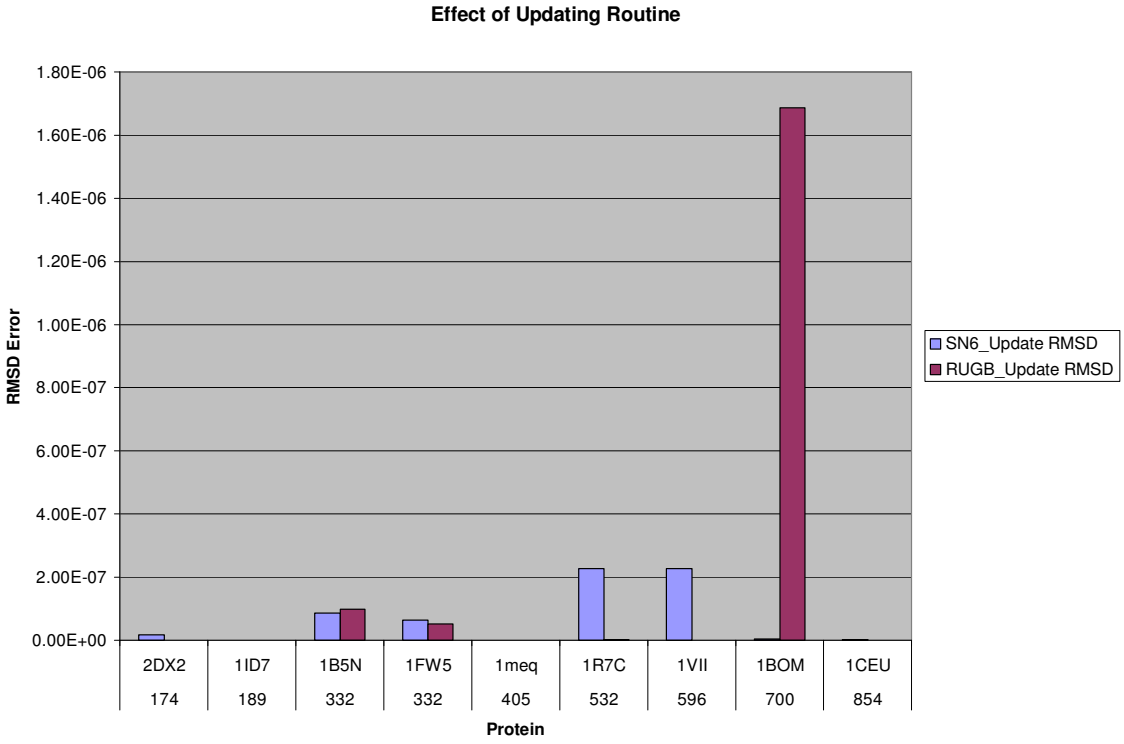


Fig. 8 Illustration of the effectiveness of the Updating routine on two differing GBU algorithms

The final graphs (Fig. 9 and Fig. 10) displayed illustrate the RMSD data for the 10 proteins comparing the new RUGB versus the SparseAlgorithm which is somewhat faster yet more unreliable and more importantly does not employ the updating routine.

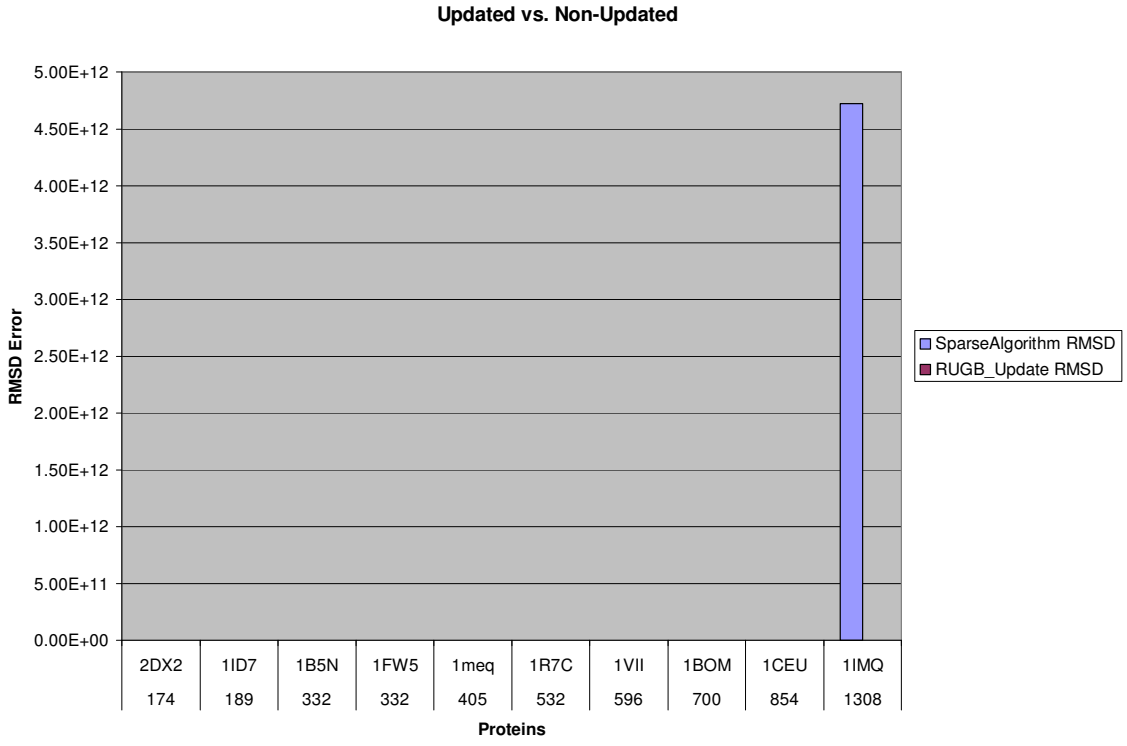


Fig. 9 Illustration of the Updating routine on 10 proteins. Comparison of SparseAlgorithm and the new Revised Updating GBU algorithm

Figure 9 shows all ten of the test proteins, which reveal an outlier that may be the result of a programming error, and Figure 10 shows only the smallest nine. This is to give a better perspective of the randomness of the error propagation in the SparseAlgorithm while also showing just how well the updating routine is able to control the error build-up.

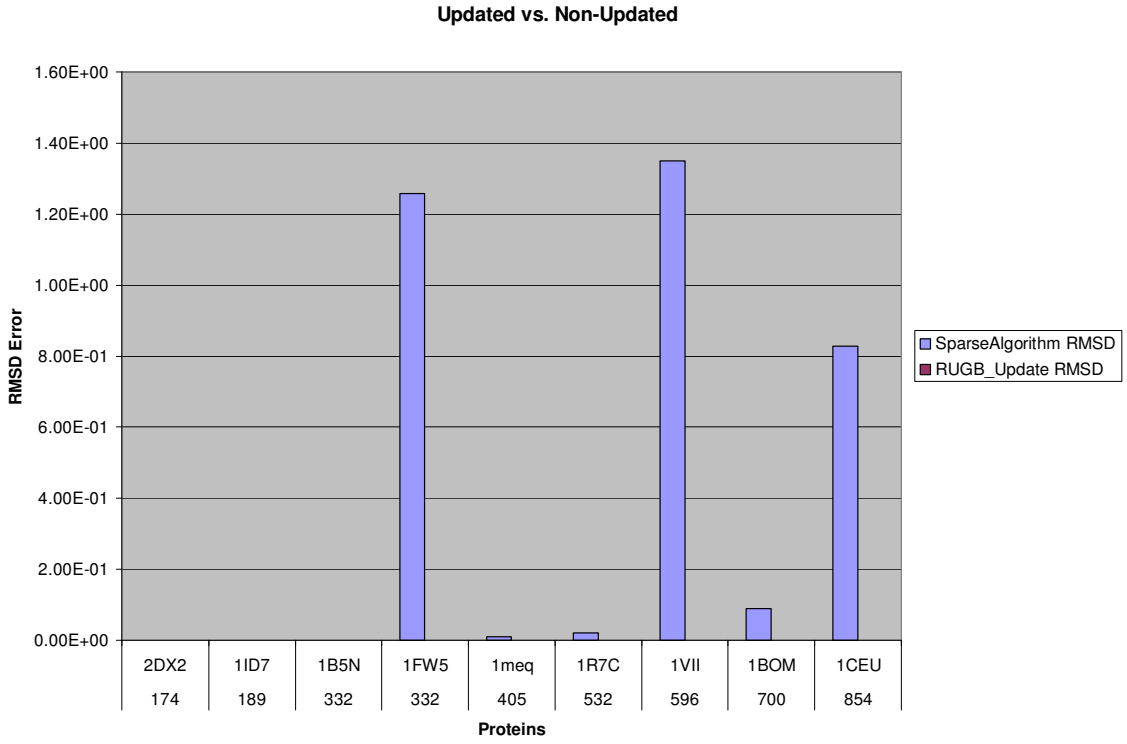


Fig. 10 Better scale for the illustration of the Updating routine on the 9 smaller proteins.

To summarize this chapter, we have outlined the evolution of the geometric build-up solution to the MDGP. Each new development has enabled the method to control error propagation, handle more sparse distance data or improve running time. We have shown a new algorithm, the RUGB, which combines many of the methods described. The following table will summarize the time and RMSD data for the ten proteins in each of the four cases.

Protein	#Atoms	SN6_Update time(s)	SN6_Update RMSD	RUGB w/o time(s)	RUGB w/o RMSD
2DX2	174	6.2882	1.70E-08	5.2595	2.24E-09
1ID7	189	7.1203	3.05E-12	2.0973	1.12E-10
1B5N	332	18.2899	8.68E-08	5.8692	2.60E-04
1FW5	332	21.2162	6.31E-08	16.7198	1.01E-08
1meq	405	27.3837	1.21E-10	6.6984	3.09E-06
1R7C	532	42.855	2.27E-07	72.847	0.2107
1VII	596	66.6627	2.27E-07	11.4302	0.0078
1BOM	700	89.5844	3.06E-09	31.1666	0.1571
1CEU	854	113.689	2.46E-09	22.5384	0.0422
1IMQ	1308	259.573	0.0845	fixable rigidly	fixable rigidly

Table 4.3
SN6_Update is an updated GBU that has a running time of $O(n^6)$
RUGB w/o is the new Revised Updated GBU without the updating routine which has a running time of less than $O(n^2 d^3_{max})$

The above running times and calculated RMSD errors can be compared to RUGB_Update data in Table 4.4 to show that we have, in fact, put together a faster and more accurate geometric build-up algorithm comprised of previous geometric build-up improvements.

Protein	#Atoms	RUGB_Update time(s)	RUGB_Update RMSD	Sparse Algorithm time	SparseAlgorithm RMSD
2DX2	174	4.9691	1.15E-11	0.1551	2.20E-05
1ID7	189	1.9283	8.00E-14	0.1562	1.21E-09
1B5N	332	4.5157	9.84E-08	1.4415	1.60E-07
1FW5	332	18.7532	5.25E-08	1.4204	1.2574
1meq	405	7.0565	5.02E-11	2.4792	0.0104
1R7C	532	56.7516	2.24E-09	6.8478	0.0213
1VII	596	12.6779	3.33E-11	8.8554	1.3509
1BOM	700	16.5135	1.69E-06	17.9107	0.0882
1CEU	854	20.1145	1.52E-10	40.4166	0.8286
1IMQ	1308	26.2067	2.05E-04	214.8065	4.72E+12

Table 4.4
RUGB_Update is the new Revised Updated GBU including the updating routine and a running time of $O(n^2 d^3_{max})$
Sparse Algorithm is a non-updating GBU algorithm for the sparse exact case

Chapter 5: Summary

5.1 Research Conclusions

In the area of protein determination the molecular distance geometry problem plays a key role. In practice, because only sparse distance ranges are available, the problem becomes very difficult. One innovative solution developed has been the geometric build-up method by Dong and Wu [1]. The GBU was first described as linear time algorithm solution for the MDGP when all exact distances are known. To better simulate the practical case the GBU was then adapted to handle sparse cases of distances. This posed yet another problem, that of accumulated error. The round-off error of earlier determined atoms being passed on to the later determined errors highly affects the accuracy of the overall structure. An updated geometric build-up solution was then described by Wu and Wu showing that the error could indeed be controlled and minimized [3]. Still, another persisting problem for the GBU is running time. Although a sparse case GBU solution [2] and a method for updating the base atoms [3] have been described the running times of $O(n^4)$ and $O(n^6)$, respectively, leave room for improvement. Until now the problem of finding the base set of atoms has not yet formally been described but has shown promise in regards to reducing the running time for the algorithm.

The work presented here describes a new Revised Updated Geometric Build-Up algorithm that not only employs the aforementioned updating routine but also introduces a novel method for more quickly finding a base set of atoms. The unique determination of an atom requires four determined atoms with all four distances to the undetermined atom. Moreover, the use of the updating routine in this situation requires the additional use of the six inter-atomic distances (a 4-clique) for the base set. The problem of finding

a 4-clique is, in itself, a very difficult problem that is likely quite costly in regards to running time. To somewhat bypass this problem the revised updated GBU algorithm searches instead for a triangle base set of only three atoms and three inter-atomic distances (a 3-clique). This can be an improvement on the previous running time for finding a 4-clique of $O(n^4)$ because a triangle can be found in just $O(d_{\max}^3)$, and in fact all triangles can be found by a brute force algorithm taking only $O(d_{\max}^2 n)$ [35].

This is done by first analyzing the distance matrix creating adjacency lists, where atoms are adjacent if the corresponding inter-atomic distance is available. These lists are bounded by the maximum number of distances available for any atom in the protein, d_{max}. These lists contain all adjacent atoms, determined and undetermined, but the base atoms must be determined. To eliminate another search or for loop bounded by the number of atoms, n, an array of ones (determined) and zeros (undetermined) is used to quickly determine that status for any atom. Then, once a triangle base set has been found the revised updated GBU uses the updating routine [3] to "correct" the base coordinates. The undetermined atom is then determined rigidly, with two reflective positions, in the independent coordinate system and then all five sets of coordinates are placed into the built-up structure. This is done by translating and rotating all five sets of coordinates, the updated triangle atoms and the two possible sets of coordinates for the undetermined atom, about the geometric centers of the two triangle base structures. At this point the fourth determined atom required for a unique determination is then found and the distances are compared so that one of the two possible sets of coordinates can be eliminated. With a running time of only $O(n^2 d_{\max}^3)$ the revised updated GBU algorithm is an improvement on the first sparse case GBU algorithm described by Dong and Wu

which had a running time of $O(n^4)$, while also including the updating routine to prevent the propagation of round-off error.

5.2 Future Directions of Study

Since first being described, the geometric build-up solution has undergone several step-wise progressions and hopefully the work presented in this thesis will prove to be another step toward a strong method for solving three-dimensional protein structures. That being said, the GBU still poses some challenging issues. In fact, the new revised updated GBU algorithm still fails to find a complete structure in some instances despite all atoms satisfying the necessary condition of four required distances. Permutations in the ordering of the atoms in the original distance matrix may affect this but how can one find an optimal ordering then becomes the question. Therefore, I would like to extend this work in the future to include these permutations. However, this problem may be sidestepped if methods can be developed so that multiple components can be "connected". By determining separate components and using distances between individual atoms or even shared atoms of the components there may be a way to construct a whole structure. Further study may also be focused on choosing the base set. The algorithm's running time may be reduced even more if a base set can be chosen such that it will have multiple undetermined atoms having all three required distances. This would eliminate the necessity of finding a base set for each undetermined atom, but will be affected by the density of the atoms in the protein most likely causing inconsistencies in its effect on the running time.

Much of the work that I propose can be dealt with via strategic programming skills. For this reason I also have hopes of continuing to improve my abilities as a programmer exploring ways to analyze the distance matrix as well as the use of different data structures enabling more pertinent information to be more readily and quickly available. Still future projects may also include a second look at the use of bond lengths, bond angles, and dihedral angles in the GBU. More generally, this thesis over the MDGP and protein determination, because it is so interdisciplinary, presents many further areas of study. As a broad, long term goal learning more in the areas of biology, chemistry, computer science, and mathematics would allow further innovation and development as well as possibly opening up new avenues for research altogether.

Bibliography

[1] Q. Dong and Z. Wu, *A linear-time algorithm for solving the molecular distance geometry problem with exact inter-atomic distances*, J. Global Optim., 22, 2002, 365-375.

[2] Q Dong, Z. Wu, *A Geometric Build-Up Algorithm for Solving the Molecular Distance Geometry Problem with Sparse Distance Data*, J. Global. Optim. 26, 2003, 321-333.

[3] D. Wu and Z. Wu, *An Updated Geometric Build-up Algorithm for solving the Molecular Distance Geometry Problem with Sparse Distance Data*, J. Global Optim., 37, 661-673.

[4] "Jöns Jacob Berzelius" in *Dictionary of Scientific Biography*, vol. 2 (1970), pp. 90–97.

[5] Hlasiwetz H. and Habermann J., Liebig's Annalem, 169, 150 (1873).

[6] D. Whitford, *Proteins: Structure and Function*, John Wiley and Sons, 2005.

[7] G.A. Petsko and D. Ringe, *Protein Structure and Function*, New Science Press Ltd, 2004.

[8] Sanger, F., *The Arrangements of Amino Acids in Proteins*. Adv. Protein Chem. 7:1-66, 1952.

[9] Anifinsen CB. Science 1973; 181:223-230. [Pubmed: 4124264].

[10] Perutz, M.F., *Crystal structure of oxyhaemoglobin*. Nature, 149, 491-496, 1942.

[11] Kendrew, J C; Bodo, G; Dintzis, H M; Parrish, R G (1958), *A three-dimensional model of the myoglobin molecule obtained by x-ray analysis*, *Nature* **181** (4610): 662–6, 1958 Mar 8, doi:10.1038/181662a0, PMID:13517261

[12] J.B. Sumner, J. Biol. Chem. 69, 435 (1926).

[13] F. Bloch, W.W. Hansen, and Martin Packard, *The Nuclear Induction Experiment*, Phys. Rev., 70, 474, 1946.

[14] M. Purcell, H.C. Torrey and R.V. Pound. *Resonance Absorptions by Nuclear Magnetic Moments in a Solid*, Phys. Rev., 69, 37, 1946.

[15] G.M. Clore, A.M. Gronenborn, *NMR of Proteins*, CRC Press, 1993.

[16] Z. Wu, *Lecture Notes on Computational Structural Biology*, World Scientific Publishing Company, 2008.

[17] G.M. Crippen and T.F. Havel, *Distance Geometry and Molecular Conformation*, John Wiley & Sons, 1988.

[18] E. Deza and M.Deza, *Dictionary of Distances*, Elsevier, 2006.

[19] L.M. Blumenthal, *Theory and Applications of Distance Geometry*, Oxford, and Clarendon Press, 1953.

[20] J.B. Saxe, *Embeddability of Weighted Graphs in k-space is Strongly NP-hard*, 17th Allerton Conf. on Communication, Control and Computing (1979).

[21] G.E. Barton Jr., R.C. Berwick, E.S. Ristad, *Computational Complexity and Natural Language*, The MIT Press, 1987.

[22] Michael Sipser (1997). *Introduction to the Theory of Computation*. PWS Publishing. ISBN 0-534-94728-X. Sections 7.3–7.5 (The Class NP, NP-completeness, Additional NP-complete Problems), pp.241–271.

[23] M. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman and Company, 1979.

[24] Paul E. Black, "NP-complete", in *Dictionary of Algorithms and Data Structures* [online], Paul E. Black, ed., U.S. National Institute of Standards and Technology. 14 August 2008. (accessed TODAY) Available from: http://www.itl.nist.gov/div897/sqg/dads/HTML/npcomplete.html

[25] B.A Hendrickson, *The Molecular Problem: Determining Conformation from Pairwise Distances* Ph.D, thesis, Cornell University, 1991.

[26] W. Glunt and T.L. Hayden and M. Raydsan, *Molecular Conformations from Distance Matrices*, J. Comput. Chem., Vol.14, No.1, pp. 114-120, 1993.

[27] J. More and Z. Wu, ε-Optimal solutions to Distance Geometry Problems via global continuation, *in Global Minimization of Non-Convex Energy Functions:Molecular conformation and Protein Folding*, P.M. Pardalos, D. Shalloway, and G. Xue, eds., American Mathematical Society, 1996, 151-168.

[28] B. Kolman and D.R. Hill, *Elementary Linear Algebra 8th Edition*, Pearson Education, Inc., 2004.

[29] G.H. Golub and C.F. van Loan, *Matrix computations, 3rd Edition*, Johns Hopkins University Press, 1996.

[30] M. Sippl and H. Sheraga, Solution of the embedding problem and decomposition of symmetric matrices, *Proc. Natl. Acad. Sci. USA*, 82, 1985, 2197-2201.

[31] D. Wu, Z. Wu, Y. Yuan, *The Solution of the Distance Geometry Problem in Protein Modeling via Geometric Build-Up*, Institute for Mathematics and its Applications.

[32] J.M. Yoon, Y. Gad, Z. Wu, *Mathematical Modeling of Protein Structure Using Distance Geometry*, 2001.

[33] T.F. Havel, Distance Geometry, in *Encyclopedia of Nuclear Magnetic Resonance*, D.M. Grant and R.K. Harris, eds., John Wiley & Sons, 1995, 1701-1710.

[34] D. Wu, Z. Wu, and Y. Yuan, *Rigid vs. Unique Determination of Protein Sttructures*, Optimization Letters, (published online, DOI: 10.1007/s11590-007-0060-7), 2007.

[35] Thomas Schank, *Algorithmic Aspects of Triangle Based Network Analysis*, Dissertation, 2007.