A Geometric Buildup Algorithm for the Molecular Distance Geometry Problem

Linear Least Squares Approximation with Divide-and-Conquer Problem Decomposition

by

Vladimir Sukhoy

A Creative Component submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Applied Mathematics

Program of Study Committee:
Zhijun Wu, Major Professor
Stephen Willson
Alexander Stoytchev

Iowa State University

Ames, Iowa

2011

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

This work contributes two new ideas to the design of the algorithms to solve the Molecular Distance Geometry Problem (MDGP) in the framework of the geometric buildup algorithm. The first idea is to determine a molecular structure more uniformly using a specific procedure for the selection of the next atom to be determined during the buildup algorithm. The second idea is to decompose the MDGP into several overlapping subproblems which are solved to get a number of overlapping clusters. These clusters are merged using optimal translation and rotation of the atoms within overlapping set between clusters. Both contributions are used to enhance a known geometric buildup algorithm with linear least squares approximation (LNLS). The resulting extended geometric buildup algorithms are evaluated on a set of benchmark problems generated from a number of known protein molecular structures. From the analysis of the solutions of the benchmark problems, both uniformness and decomposition result in improvement of the algorithm's performance. Because the two proposed algorithmic enhancements are motivated by two very general principles: the Occam's razor and the divide-and-conquer paradigm, these enhancements may improve the performance of any geometric buildup algorithm.

CHAPTER 1. Introduction

This work contributes to the study of the Molecular Distance Geometry Problem (MDGP). The problem arises in the field of bioinformatics when it is necessary to determine the three-dimensional structure of a molecule given a number of inter-atomic distance constraints. These constraints emerge from theoretical models of the inter-atomic bonds, statistical analysis of the known molecular structures or Nuclear Magnetic Resonance (NMR) spectroscopy.

Let n be the number of atoms in the molecule. Let $x_1, x_2, \ldots, x_n$ be the coordinate vectors of the atoms, so that $x_i = (x_{i,1}, x_{i,2}, x_{i,3})^T$, where $x_{i,1}, x_{i,2}, x_{i,3}$ are the numerical coordinates of the i-th atom. Let S be the set of pairs of atom indices for which a distance constraint is known. Solving the MDGP is the act of finding the coordinate vectors $x_1, \ldots, x_n$ given n, S, and the distance constraints. Young and Householder (Y38) gave the necessary and sufficient conditions for a set of distances to define a set of points in Euclidean space that satisfy these conditions. During the solution process, an atom is labeled as *determined* if its coordinates are already found, otherwise it is labeled as *undetermined*.

Depending on the type of distance constraints, the MDGP can be *exact* or *inexact*. Let $d_{i,j}$ denote the Euclidean distance between the i-th and j-th atom, i.e.,

$$d_{i,j} = \sqrt{(x_{i,1} - x_{j,1})^2 + (x_{i,2} - x_{j,2})^2 + (x_{i,3} - x_{j,3})^2}.$$

If the value of $d_{i,j}$ is known exactly for all $(i,j) \in S$, then the MDGP is *exact*. If at least some of the distance constraints are provided in another form, then the MDGP is *inexact*. One important type of inexact MDGP that arises in practice is the MDGP with distance range constraints, i.e., when these constraints are in the form $l_{i,j} \leq d_{i,j} \leq u_{i,j}$ for $(i,j) \in S$.

Solving the MDGP contributes to the study of proteins – an important class of biological molecules that participate in numerous processess within living organisms. The proteins are

polypeptides – chains of amino acid molecules that are connected by peptide bonds to form an aggregate polypeptide molecule. Proteins are constructed in cells through genetic transcription – a process of interpreting the genetic code. After the transcription, the protein molecule folds into a stable three-dimensional structure. During folding, the spatial structure of the molecule changes as the atoms within the molecule interact with each other and with the atoms that surround the protein molecule. The duration of folding depends on the particular protein and varies from microseconds to hours. Protein folding is challenging to model as a computational process. Yet, the three-dimensional structure of a folded protein is key to understanding its function, i.e., what it does inside a living organism. There are two experimental approaches for determining the spatial structure of a folded protein: X-Ray crystallography and NMR spectroscopy. To apply X-Ray crystallography, a protein must be crystallized, however, which is not possible for many important proteins. On the other hand, NMR spectroscopy can work with protein molecules in a solution. The output of an NMR spectroscopy experiment can be used to generate a number of inter-atomic distance constraints. These constraints, together with statistics on the known structures, can be used to formulate an instance of the MDGP. The three-dimensional structure of the protein molecule is the solution of this MDGP instance.

In general, the MDGP was proven to be NP-hard (S79). That is, the time required to solve an instance of the MDGP may grow exponentially with n. Yet, this is often not the case for the MDGP instances encountered in practice. Increasing the number of the distance constraints for an exact MDGP instance with fixed n may reduce its computational complexity. In particular, an exact MDGP instance where all $\binom{n}{2}$ distances are available can be solved in polynomial time by factorizing a matrix generated from the distance constraints, e.g., using Singular Value Decomposition (SVD).

The MDGP is a special case of a more general problem: to characterize points in a topological space given the distance constraints for the certain pairs of these points. In Mathematics, this more general problem is called the *distance geometry problem* (B70). This problem may be formulated in any metric space (B70), i.e., it is not restricted to $\mathbb{R}^3$ with Euclidean distances. In Computer Science, a closely related problem is the *graph embedding problem* (S79). The

solution for the graph embedding problem is a representation for a weighted graph within a chosen metric space that preserves the structure of the graph. In Statistics, Multidimensional Scaling is a set of techniques for solving various problem that can be related to distance geometry problem in Mathematics (T58). This work focuses exclusively on the MDGP, i.e., the distance geometry problem is formulated in $\mathbb{R}^3$ with Euclidean distances.

This work contributes to solving the MDGP within the framework of geometric buildup algorithms (SWY09). Within the core of this framework lies the idea that the MDGP can be solved iteratively. The position of one atom is determined at each iteration of a geometric buildup algorithm. The geometric relationships between the atoms that are already determined and the constraints on the distances between these atoms the atom to be determined at the current iteration are used to formulate a system of equations. The solution to this system is a vector with the coordinates of the atom being determined. The geometric requirements are: the number of already determined atoms, which are used to formulate the system, must be greater than three and these atoms must not lie on the same plane. Furthermore, all constraints on the distances between the already determined atoms and the atom to be determined must be available. Despite the progress in recent work (WW07) (SWY09), sensitivity to rounding error and error in the distance constraints, propagation and gain of these errors through the iterations remain major issues for the algorithms in the framework of the geometric buildup. The contributions of this work enhance the framework so that these issues are less prominent.

This work proposes a heuristic to choose an atom to be determined at each iteration of the algorithm. The heuristic improves the uniformness of intermediate structures. When the heuristic is applied, an atom is determined at the earliest possible iteration. Doing so reduces the ability of error to propagate and increase. This work also proposes the decomposition procedure for the MDGP. The problem instance is decomposed into several overlapping subproblems. These subproblems are solved to obtain clusters of determined atoms. Each of these clusters is a part of the solution to the original MDGP instance before decomposition. Each of the clusters is determined in the coordinate system local to the particular cluster. Clusters are then merged using coordinate transformations computed from the coordinates of the atoms

in the overlap. The result of merging is the solution to the original MDGP. Decomposition reduces the size of each subproblem to solve with a geometric buildup algorithm. This reduction also limits the error in each individual cluster. It is expected that merging of the clusters introduced less error than equivalent amount of geometric buildup iterations because clusters typically overlap by more atoms than are used during an iteration.

Both contributions of this work are evaluated on a set of benchmark MDGP instances that have been used previously in literature. The evaluation shows that geometric buildup algorithms that benefit from the contributions described here produce better results than the analogues of the algorithms that do not benefit from the contributions.

CHAPTER 2. Related Work

The MDGP is a special case of the graph embedding problem in computer science. Saxe (S79) proved several results on computational complexity of the graph embedding problem. These results, which carry over to the MDGP case, can be summarized as follows:

- In general, a one-dimensional graph embedding problem with exact integer weights is NP-Complete and remains NP-Complete even if edge weights are guaranteed to be bounded by a polynomial on the number of edges. For the exact MDGP case, this means that if it were formulated in $\mathbb{R}^1$ and the distance were restricted to integers, the resulting problem is NP-Complete.

- The graph embedding problem in $\mathbb{R}^k$, where k is a positive integer is strongly NP-hard.

The above results imply that the exact MDGP is strongly NP-hard in $\mathbb{R}^k$.

Moré and Wu (MW95) showed that the approximate distance geometry problem where the solution is allowed to deviate from the distance constraints is NP-hard if the errors are small. Wu et al. (WWY07) give another proof of NP-hardness of the approximate distance geometry problem that only requires the sum of allowed errors to be bounded by a constant. For the special case of MDGP, these results imply that even a relaxed version of the exact MDGP where the distances may violate the constraints is, in general, NP-hard as long as there is a bound on the sum of all errors.

Crippen and Havel (GH88) proposed the EMBED algorithm for solving the distance geometry problem in mathematics. This algorithm can be applied to solve the MDGP, as it is simply a special case of the distance geometry problem. The idea of the EMBED algorithm, in terms of the MDGP, is to estimate all pairwise distances between the atoms from the available

constraints so that the resulting set of distance constraints is geometrically consistent. Once all distances between the atoms are obtained, the approximate solution to the original problem can be obtained using distance matrix factorization (see Chapter 3). The EMBED algorithm further refines this solution using local optimization: it reduces the error between the distances computed from the approximate solution and the distance constraints.

Hendrickson (H92) formulated three necessary conditions for the unique solvability of the graph realization problem in $\mathbb{R}^k$. The MDGP is a special case of the graph realization problem in $\mathbb{R}^3$ with the target structures being biological molecules. In terms of the graph realization problem, the exact MDGP's set of constraints is a weighted graph where the vertices are the MDGP's atoms and the edge weights are the distance values. The goal of the MDGP, in terms of the graph realization problem, is to realize the graph in $\mathbb{R}^3$. Hendrickson (H92) proved that for the graph to be uniquely realizable it must be rigid, $(k+1)$-connected, and redundantly rigid. Hendrickson (H95) also proved the sufficient condition for solvability of the graph realization problem, where this sufficient condition is formulated in the framework of stress matrices. Hendrickson (H95) proposed the ABBIE divide-and-conquer algorithm to solve the graph realization problem. The algorithm divides the graph into uniquely realizable subgraphs. The division process is guided by the necessary and sufficient conditions. The algorithm uses a global optimization method to solve the problem for each subgraph and merges the results to obtain the realization of the whole graph. The idea of decomposition in ABBIE is similar to the decomposition introduced in this work. However, primary goal and the criteria for decomposition are different. In this work the criterion is formulated in the framework of the geometric buildup algorithm and the goal is to improve the quality of algorithm's output. In ABBIE the criterion is to ensure theoretical solvability of the subproblems and the goal is to reduce time and computational effort.

Biswas et al. (BLTY05), (BTY08) combined semidefinite programming, problem decomposition, and gradient descent to solve the MDGP. They formulated the problem in the framework of the graph realization problem. Semidefinite programming and problem decomposition provide the trial solution that is further refined using gradient descent. The need to reduce

the computational effort of semidefinite programming motivates the decomposition process (BTY08). The clusters from the problem decomposition are stitched together using optimal rotation and translation of the atoms in the cluster overlap. The stitching is similar to the merging procedure that combines clusters in this work.

Zou et al. (ZBS97) proposed a stochastic/perturbation global optimization algorithm to solve the MDGP. The algorithm seeks to find the minimum of the function that penalizes all unsatisfied distance constraints in the structure. The algorithm consists of two sequential phases: a stochastic phase followed by a deterministic phase. The stochastic phase combines random sampling of the MDGP's domain with local optimizations to generate molecular structures. Each of these structures locally minimizes the penalty function. The deterministic phase iteratively refines these locally minimal structures by global and local optimization. The algorithm returns the refined structure with the lowest value of the penalty function.

Trosset (T98) sketched the mathematical formluation for using multidimensional scaling (MDS) for solving the MDGP. In particular, two approaches were described: 1) the approach based on the Stress criterion for metric MDS; 2) the approach based on Strain criterion for classical MDS. The first approach, combined with the Spectral Gradient Method, is the basis for the Data Box Algorithm (GHR93).

2.1 Previous Work on Geometric Buildup

Sippl and Scheraga (SS85), (SS86) were the first to propose the geometric buildup approach for the alternative formulation of the distance geometry problem in the framework of a matrix completion problem. Later, Dong and Wu (DW02) proposed the geometric buildup as an alternative algorithm for solving the exact MDGP when all distances are available. The algorithm is based on a simple geometric fact in 3D Euclidean space: if the coordinates of four points that do not lie on the same plane are known, then the coordinates of an arbitrary point can be found given the distances from that point to the four known points. The four resulting distance equations can be reduced to a linear system that can be solved in constant time. It follows that the algorithm requires only $4n$ distances and only $O(n)$ operations to complete (unlike

the approach based on matrix factorization described in Chapter 3, which finishes in quadratic time).

Dong and Wu (DW03) also extended the algorithm to handle problems with a sparse set of available distances. Unfortunately, the resulting algorithm was sensitive to a numerical error that propagated and increased throughout the computation process. Control over the numerical error was the major goal of the later work on geometric buildup. Wu and Wu (WW07) proposed an updating scheme that significantly reduced the propagation of numerical error by re-evaluating the coordinates of any four determined atoms whenever possible. The recalculated coordinates do not inherit the error from the previous stages of the algorithm because they do not depend on the coordinates of any previously determined atoms.

Sit, Wu, and Yuan (SWY09) proposed several further ideas to improve the ability of the algorithm to handle the numerical error and to tolerate small errors in the available distances. First, when determining an atom all available distances are used (instead of a subset of a fixed size), thus forming a possibly overdetermined system of distance equations that can be solved using least-squares approximation. Second, the nonlinear least squares formulation of the approximation problem was given and a solution method using SVD that limits the numerical error accumulation was provided. The more traditional linear least-squares formulation of the approximation problem from (SWY09) is used as a step of the algorithm described in this work due to its lower computational requirements compared to the SVD method. This work adds uniformness and problem decomposition into clusters to the linear least-squares approximation procedure.

CHAPTER 3. Background

3.1 Solving the MDGP with a full set of distances

The goal of this section is to prove that the exact MDGP (i.e., the MDGP with exact distance constraints) with all $\binom{n}{2}$ distance constraints can be solved in polynomial time. The proof is restructed to the MDGP in $\mathbb{R}^3$. For a proof in $\mathbb{R}^k$ for any $k \in \mathbb{N}$, which can be also applied for the case when $k = 3$, see (WWY07).

Each of the $\binom{n}{2}$ distance constraints can be written as follows:

$$d_{i,j}^2 = ||x_i||^2 - 2x_i^T x_j + ||x_j||^2 , \tag{3.1}$$

where $i = 1, \ldots, n-1$, and $j = i+1, \ldots, n$. Because the inter-atomic distances are invariant under translations, it is possible to set $x_n = 0$ and$||x_i||^2 = d_{i,n}^2$ without loss of generality. Replacing $||x_i||^2$ with $d_{i,n}^2$ allows us to rewrite the distance constraints (3.1) as shown below:

$$d_{i,j}^2 = d_{i,n}^2 - 2x_i^T x_j + d_{j,n}^2. \tag{3.2}$$

Let $X = [x_i]$, where $i = 1, \ldots, n$, be the $n \times 3$ matrix of atom coordinates, i.e.,

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

Also let $D = \frac{1}{2}\left[d_{i,n}^2 - d_{i,j}^2 + d_{j,n}^2\right]$, where $i, j = 1, \ldots, n$, be the $n \times n$ matrix that represents all

distance constraints, i.e.,

$$D = \frac{1}{2}\begin{bmatrix} \left(d_{1,n}^2 - d_{1,1}^2 + d_{1,n}^2\right) & \left(d_{1,n}^2 - d_{1,2}^2 + d_{2,n}^2\right) & \cdots & \left(d_{1,n}^2 - d_{1,n}^2 + d_{n,n}^2\right) \\ \left(d_{2,n}^2 - d_{2,1}^2 + d_{1,n}^2\right) & \left(d_{2,n}^2 - d_{2,2}^2 + d_{2,n}^2\right) & \cdots & \left(d_{2,n}^2 - d_{2,n}^2 + d_{n,n}^2\right) \\ \vdots & \vdots & \cdots & \vdots \\ \left(d_{n,n}^2 - d_{n,1}^2 + d_{n,n}^2\right) & \left(d_{n,n}^2 - d_{n,2}^2 + d_{2,n}^2\right) & \cdots & \left(d_{n,n}^2 - d_{n,n}^2 + d_{n,n}^2\right) \end{bmatrix}.$$

Note that the equations for distance constraints (3.2) can be rewritten in the following form:

$$x_i^T x_j = \frac{1}{2}\left(d_{i,n}^2 - d_{i,j}^2 + d_{j,n}^2\right), \qquad (i,j = 1,\ldots,n).$$

The above set of n^2 equations can be also written in matrix form in terms of the coordinate matrix X and the matrix D as shown below:

$$X^T X = D. \tag{3.3}$$

Note that, because the rank of the matrix X is at most 3, the rank of D is also at most 3.

Let $D = U\Sigma V^*$ be the *singular value decomposition* (SVD) of the matrix D (GL89). Because the matrix D is symmetric, $V^* = U^T$. In other words, the SVD of the matrix D can be expressed as follows:

$$D = U\Sigma U^T, \tag{3.4}$$

where U is a unitary matrix and Σ is a nonnegative diagonal matrix. Given the SVD of the matrix D (3.4), it is possible to compute a suitable coordinate matrix X, which solves the MDGP, as follows:

$$X = U\Sigma^{\frac{1}{2}}. \tag{3.5}$$

Becasue the rank of the matrix D is at most 3, its SVD (3.4) can be computed in $O(n^2)$ operations (GL89). Furthermore, the number of operations required to compute the coordinates matrix X using (3.5) is also $O(n^2)$. It follows that the overall computational cost of solving the exact MDGP with a full set of distance constraints is $O(n^2)$ operations.

3.2 Evaluating Solutions to the MDGP Using RMSD, DME, LDME

Once an instance of the MDGP is solved, it is necessary to evaluate how good the solution is. If the problem was generated for a known molecule, the structure of a solution can be compared with the structure of the molecule. For the root-mean-square deviation (RMSD), the coordinates of the atoms in both structures are compared directly. For the distance matrix error (DME), the pairwise distances between the atoms in the two structures are compared. If the parent molecule is not known, the solution is good if it satisfies the constraints of the MDGP. For the exact MDGP, the local distance matrix error (LDME) is a measure of how well the solution satisfies the constraints of the problem.

The RMSD is frequently used in bioinformatics to compare spatial molecular structures (MC94). In essence, the RMSD is equal to the numerical value of the Frobineus norm of the difference between the coordinate matrices of the two structures given that the coordinate matrices are translated and rotated to match as closely as possible.

Formally, let $X = \left[x^{(1)}, \ldots, x^{(n)}\right]^T$ be the $n \times 3$ matrix of the solution coordinates and let $Y = \left[y^{(1)}, \ldots, y^{(1)}\right]$ be such a matrix for the parent structure. First, the matrices are translated so that they are centered at the origin

$$\begin{aligned} X' =& X - \frac{1}{n}\sum_{i=1}^{n} x^{(i)}, \\ Y' =& Y - \frac{1}{n}\sum_{i=1}^{n} y^{(i)}. \end{aligned}$$

The RMSD is the scaled value of the Frobineus norm of the difference between optimally oriented structures:

$$\text{RMSD} = \sqrt{\frac{1}{n}}\left|\left|QX' - Y'\right|\right|_F,$$

where Q is a 3×3 orthonormal matrix that orients the solution optimally with respect to the parent structure.

Computing the optimal orientation matrix Q for the two structures is an important step in the algorithm developed in this work (see Figure 5.2). This can be done using Singular Value Decomposition: if $U\Sigma V^* = \text{svd}(Y^*X)$ then $Q = UV^*$ (K76).

Another way to evaluate the quality of the solution is to compare the distances in the solution structure against the distances in the parent structure or against the known distances. The mean error of all distances is called the distance matrix error and is computed as follows:

$$\text{DME} = \sqrt{\binom{2}{N}^{-1} \sum_{i=2}^{n} \sum_{j=1}^{i-1} \left(\left|\left| x^{(i)} - x^{(j)} \right|\right| - \left|\left| y^{(i)} - y^{(j)} \right|\right| \right)^2}.$$

The mean error for only those distances that are known in the problem is called the local distance matrix error:

$$\text{LDME} = \sqrt{\frac{1}{|\mathcal{D}|} \sum_{(i,j)\in\mathcal{D}} \left(\left|\left| x^{(i)} - x^{(j)} \right|\right| - d_{i,j} \right)^2}.$$

3.3 The General Geometric Buildup Approach

In general, the geometric buildup algorithm starts by finding the first four atoms to be determined so that all pairwise distances between them are available and they are do not lie in the same plane. Given these distances, the Cayley-Menger determinant can be used to check if the atoms lie in the same plane. Once these first four atoms are found, their coordinates can be assigned either from geometric relationships or using the above method based on distance matrix factorization. The atoms are then fully determined.

To determine the remaining atoms, the algorithm iterates as follows: it repeatedly seeks an undetermined atom j so that there are four determined atoms that do not lie in the same plane and the distances between these atoms and atom j are known. Then, the coordinates of atom j are determined using the pairwise distances and the coordinates of the four atoms. Formally, without loss of generality, let $j > 4$ and let the coordinates of the for determined atoms be $x_i = (x_{i,1}, x_{i,2}, x_{i,3})^T$, where $i = 1, 2, 3, 4$. Then, x_j can be found from the system of distance equations

$$||x_i - x_j|| = d_{i,j},\ i = 1, 2, 3, 4.$$

Since these distances are Euclidean distances, the system can be written as follows:

$$||x_i||^2 - 2x_i^T x_j + ||x_j||^2 = d_{i,j}.$$

The above system is quadratic, but it becomes a linear system if the $(i+1)$-th equation is subtracted from the i-th equation:

$$2\left(x_i - x_{i+1}\right)^T x_j = \left(||x_{i+1}||^2 - ||x_i||^2\right) - \left(d_{i+1,j}^2 - d_{i,j}^2\right),\ i = 1, 2, 3.$$

Let A be a 3×3 matrix and let b be a 3×1 vector, which are defined as follows:

$$A = 2\begin{bmatrix} \left(x_1 - x_2\right)^T \\ \left(x_2 - x_3\right)^T \\ \left(x_3 - x_4\right)^T \end{bmatrix}, \quad b = \begin{bmatrix} \left(||x_2||^2 - ||x_1||^2\right) - \left(d_{2,j}^2 - d_{1,j}^2\right) \\ \left(||x_3||^2 - ||x_2||^2\right) - \left(d_{3,j}^2 - d_{2,j}^2\right) \\ \left(||x_4||^2 - ||x_3||^2\right) - \left(d_{4,j}^2 - d_{3,j}^2\right) \end{bmatrix}.$$

The linear system that determines x_j is $Ax_j = b$. The solution always exists because A is nonsingular (if A is singular then x_1, x_2, x_3, x_4 must be on the same plane, which is not the case). Solving a linear system of a constant size requires a constant amount of computation. If, for each j, the distances $d_{i,j}$, $i = 1, 2, 3, 4$ are known, then the algorithm (see Fig. 3.1) only requires solving $n - 4$ linear systems plus the determination of the first four atoms. In this case, the computational complexity of the generic geometric buildup algorithm is $O(n)$.

Input: The set S of atom pairs, for which the distances are available;
Find four atoms that are not on the same plane such that all pairwise distances between these atoms are available;
Determine the coordinates of the atoms using the pairwise distances;
repeat
  foreach *undetermined atom* **do**
    if *distances between the undetermined atom and four determined atoms are available* ***and*** *these determined atoms are not on the same plane* **then**
      Determine the atom by solving the linear system specified by the distance equations;
    end
  end
until *no new atoms were determined* ;

Figure 3.1 General Geometric Buildup Algorithm

The theoretic basis for the geometric buildup approach builds on the concepts of *metric basis* and *set of independent points* that are used in distance geometry theory (B70).

Definition. A set of points B in a metric space is a metric basis if the coordinates of each point in the space can be uniquely determined from the distances between that point and the

other points in B.

Definition. A set of $k+1$ points in $\mathbb{R}^k$ is a set of independent points if it cannot be embedded in $\mathbb{R}^{k-1}$.

The following theorem generalizes the geometric buildup iteration in an arbitrary finite-dimensional Euclidean space.

Theorem. *Any $k+1$ independent points in $\mathbb{R}^k$ form a metric basis for $\mathbb{R}^k$.*

Proof. The above $\mathbb{R}^3$ justification for the iteration of the general geometric buildup algorithm (see Figure 3.1) is straightforward to reformulate in $\mathbb{R}^k$. Let $x_i = (x_{i,1}, \ldots, x_{i,k})^T$ for $i = 1, \ldots, k+1$ be the coordinate vectors of a set of independent points in $\mathbb{R}^k$, $y = (y_1, \ldots, y_k)^T$ be the coordinates of an arbitrary point y in $\mathbb{R}^k$, and d_i be the Euclidean distance from x_i to y. Then $||x_i||^2 - 2x_i^T y + ||y||^2 = d_i$ for $i = 1, \ldots, k+1$. Letting

$$A = 2\begin{bmatrix} (x_1 - x_2)^T \\ (x_2 - x_3)^T \\ \vdots \\ (x_k - x_{k+1})^T \end{bmatrix}, \quad b = \begin{bmatrix} \left(||x_2||^2 - ||x_1||^2\right) - \left(d_2^2 - d_1^2\right) \\ \left(||x_3||^2 - ||x_2||^2\right) - \left(d_3^2 - d_2^2\right) \\ \vdots \\ \left(||x_{k+1}||^2 - ||x_k||^2\right) - \left|\left|d_{k+1}^2 - d_k^2\right|\right| \end{bmatrix}$$

we see that $Ay = b$. This linear system must have a solution because A is nonsingular for otherwise the set of points $\{x_1, \ldots x_{k+1}\}$ can be embedded in $\mathbb{R}^{k-1}$. □

3.4 Linear Least Squares Approximation for the Geometric Buildup Algorithm

The general geometric buildup algorithm (see Figure 3.1) only uses four distances when determining the new atom. This implies that if there are more than four distances available from the determined atoms to the one being determined, any additional distances will be excluded from the computation. Given the possibility of error in the distances and the numerical error in the computations, it is desirable to use all available distances in each step of the geometric

buildup. (SWY09) proposed using linear least squares to find the most optimal solution to the overdetermined system of distance equations.

Formally, including all, say $l \geq 4$ available distances to determine the $j-th$ atom in the general geometric buildup step replaces A and b with

$$A = 2\begin{bmatrix} (x_1 - x_2)^T \\ (x_2 - x_3)^T \\ \vdots \\ (x_{l-1} - x_l)^T \end{bmatrix}, \quad b = \begin{bmatrix} \left(||x_2||^2 - ||x_1||^2\right) - \left(d_{2,j}^2 - d_{1,j}^2\right) \\ \left(||x_3||^2 - ||x_2||^2\right) - \left(d_{3,j}^2 - d_{2,j}^2\right) \\ \vdots \\ \left(||x_l||^2 - ||x_{l-1}||^2\right) - \left(d_{l,j}^2 - d_{l-1,j}^2\right) \end{bmatrix}.$$

The resulting (possibly) overdetermined system $Ax_j = b$ can be approximated using least-squares algorithm, e.g., via QR factorization.

Input: The set S of atom pairs, for which the distances between them are available;
Find four atoms that are not on the same plane such that all pairwise distances are available;
Determine the coordinates of the atoms using the pairwise distances;
repeat
  foreach *undetermined atom* **do**
    if *distances between the undetermined atom and four determined atoms are available* ***and*** *these determined atoms are not on the same plane* **then**
      Determine the atom by approximating the solution to a possibly overdetermined linear system of equations that follows from distance constraints using the least squares method;
    end
  end
until *no new atoms were determined* ;

Figure 3.2 Geometric Buildup Algorithm using Linear Least Squares Approximation

The algorithm developed in this work uses the above linear least-squares procedure for determine the next atom. However, in contrast to the linear least-squares algorithm in (SWY09), the choice of the atom to determine next is made using a heuristic that ensures the uniform growth of the set of determined atoms with respect to the first four atoms.

CHAPTER 4. Methodology

Two principles guided the design of algorithms proposed in this work. The first principle favors algorithms that produce a simpler solution. The second principle augments a geometric buildup algorithm with a scheme that decomposes a larger MDGP into MDGP sub-instances and merges their solutions to obtain the solution to the original MDGP.

The design principle of *uniformness* is an application of the general meta-theoretic principle that favors simpler solutions over more complex ones (Sto01). In the context of the geometric buildup framework, this principle suggests that an algorithm that produces simpler solutions is preferrable to one that produces more complicated solutions. To quantify the complexity of a solution, the *generation index* is computed for each of the atoms as they are determined. This computation can augment any algorithm in the GB framework without modifying the selection rule by which atoms are determined at each iteration of the algorithm (see 4.1). For a given atom a, the generation index measures how many "layers" of atoms had to be determined before it. The maximum generation index over all atoms in the solution is a numerical measure of its complexity.

To seek a simpler solution, an algorithm can minimize its numerical measure, i.e., the maximum generation index over all determined atoms. Thus, the structure is determined in fewer generations and the number of atoms in each generation is maximized, making the process of solving the MDGP more uniform. An algorithm can adopt a greedy approach and always select an atom a in line 9 of Fig. 4.1 for which $G[a]$ is minimized. The result is the *uniform* general geometric buildup algorithm (see Fig. 4.2), which always chooses the next atom using greedy optimization with respect to the design principle of uniformness.

The decomposition scheme described above is an application of the divide-and-conquer

Input: The set S of atom pairs, for which the pairwise distances are available;
Output: The coordinate matrix X of the determined atoms;
Output: The generation indices G for the determined atoms;
$G \leftarrow$ empty list of atom generation indices;
Find four atoms a_1, a_2, a_3, a_4 for which all 7 pairwise distances between them are available and the atoms are not on the same plane;
`// initialize the generation indices for the first four atoms`
$(G[a_1], G[a_2], G[a_3], G[a_4]) \leftarrow (0, 0, 0, 0)$;
Determine the coordinates of the first four atoms $X[a_1]$, $X[a_2]$, $X[a_3]$, $X[a_4]$ using the pairwise distances;
repeat
  foreach *undetermined atom* a **do**
    if *at least four distances between* a *and determined atoms* $b_1, \ldots, b_k$, $k \geq 4$ *are available* ***and*** *these determined atoms are not on the same plane* **then**
      Determine the coordinates of a using the coordinates and distances between a and $b_1, \ldots, b_k$;
      $G[a] \leftarrow 1 + \max_{i=1,\ldots,k} G[b_i]$; `// set the generation index of` a
    end
  end
until *no new atoms were determined* ;

Figure 4.1 General geometric buildup algorithm enhanced with computation of the atom generation index.

Input: The set S of atom pairs, for which pairwise distances are available;
Output: The coordinate matrix X for the determined atoms;
Find four atoms a_1, a_2, a_3, a_4 for which all pairwise distances are available and which do not lie on the same plane;
`// initialize the generation indices for these first four atoms`
$(G[a_1], G[a_2], G[a_3], G[a_4]) \leftarrow (0, 0, 0, 0)$;
Determine the coordinates $X[a_1]$, $X[a_2]$, $X[a_3]$, $X[a_4]$ using the pairwise distances between a_1, a_2, a_3, a_4;
repeat
  $N \leftarrow$ the set of all undetermined atoms, for each of which there are ≥ 4 adjacent atoms in S and these neighbors are determined and not on the same plane;
  $a \leftarrow \arg\min_{a \in N} \max_{\{a,b\} \in S, b \text{ is determined}} G[b]$;
  Determine the coordinates of atom a using the coordinates of (and distances to) the determined adjacent atoms in S;
  $G[a] \leftarrow 1 + \max_{\{a,b\} \in S, b \text{ is determined}} G[b]$;
until *no new atoms were determined* ;

Figure 4.2 Uniform general geometric buildup algorithm applying greedy approach to select the next atom to be determined.

algorithm design paradigm (Knuth98). Under this paradigm, a problem is broken down into several sub-problems until they become small enough to be solved directly. In the context of the geometric buildup framework, this paradigm leads to a scheme (see Fig. 4.3) that decomposes the MDGP into sub-problems of sufficiently small size, such that they can be efficiently solved by a geometric buildup algorithm to obtain *clusters* of determined atoms, each of which is solved in its local coordinate system. The maximum generation index of the determined atoms in a cluster quantifies the complexity of a cluster. A limit on this index is a limit on the complexity of each individual sub-problem. To combine the solutions, the scheme requires these clusters to overlap. The scheme uses the set of overlapping atoms to find the optimal transformation between the coordinates of these atoms in both sub-structures. The scheme computes the transformation using the same computations that are used to find the RMSD of two structures (see Section 3.2). Then the scheme applies the transformation to all atoms in one of the two sub-structures and merges them with the coordinates of the atoms in the other sub-structure. The scheme is applicable to any GB algorithm provided that it is enhanced with computation of the atom generation index.

This work proposes several new approaches to solving the MDGP within the geometric buildup framework (DW02). Achieving numerical stability of the algorithms in the framework remains a challenge, despite the progress reported in (WW07) and (SWY09). The ability of the framework to solve MDGP instances with inexact or approximate distance constraints needs to be improved for application to practical MDGP problems derived from NMR experiments.

Input: The set S of atom pairs, for which the distances are available;
Input: The limit L on the maximum generation index over atoms in each cluster;
Output: The list of clusters $\mathcal{C}$ of all determined atoms;
$\mathcal{C} \leftarrow$ empty list of atom clusters;
`// generate clusters`
while *can find four atoms* a_1, a_2, a_3, a_4, *for which all pairwise distances are available* ***and*** *these atoms are not on the same plane* ***and*** *at least one of these atoms is not determined in each cluster in* $\mathcal{C}$ **do**
    Invoke the geometric buildup algorithm iteration starting from a_1, a_2, a_3, a_4 until the maximum generation index is $\leq L$. The result is the atom cluster C in its local coordinate system;
    Insert C into $\mathcal{C}$;
end
`// merge clusters`
repeat
    foreach $C_1 \in \mathcal{C}$, $C_2 \in \mathcal{C}$ *that overlap by* ≥ 4 *atoms* ***and*** *these atoms are not on the same plane* **do**
        Merge C_1 and C_2 using optimal translation and rotation of the set of overlapping atoms into a combined cluster C;
        Remove C_1 and C_2 from $\mathcal{C}$;
        Insert C into $\mathcal{C}$;
    end
until *no two clusters in* $\mathcal{C}$ *overlap by* ≥ 4 *atoms* ***and*** *these atoms are not on the same plane* ;

Figure 4.3 Problem decomposition scheme applicable to any GB algorithm, provided that it computes generation indices for derived atoms as shown in Fig. 4.1.

CHAPTER 5. Results

The algorithms proposed in this work use two general design principles described in Chapter 4: the principle of *uniformness* and the divide-and-conquer *decomposition* scheme. These principles are applied to the design of the geometric buildup algorithm with linear least squares approximation (LNLS algorithm) from (SWY09), shown in Fig. 3.2.

First, the LNLS algorithm is enhanced with computation of generation indices for the newly determined atoms. Second, the order of atom determination is modified to select the atom to be determined next such that the generation index is minimized over the set of all atoms that can be determined next. The result is an implementation of the LNLS algorithm with respect to the design principle of uniformness, i.e., the result is the *uniform* geometric buildup algorithm with linear least squares approximation (the ULNLS algorithm, see Fig. 5.1). Finally, the decomposition scheme is applied to the ULNLS algorithm. The result is the *uniform* geometric buildup algorithm using linear least squares approximation enhanced with the *problem decomposition* scheme (the ULNLSPD algorithm, see Fig. 5.2). By setting the limit L on the maximum generation index in each cluster of determined atoms to infinity, the decomposition scheme in the ULNLSPD algorithm can be disabled, effectively transforming it into the ULNLS algorithm.

The performance of both ULNLS and ULNLSPD was evaluated on a benchmark of MDGP instances. These instances were used in literature (SWY09) (WW07) (GLS09) (BLTY05) (DW02) to evaluate different algorithms for solving the MDGP. Each of these instances is generated from a known structure in the PDB databank (BHN03) by selecting only those inter-atomic distances that do not exceed a specified cutoff value (which is typically set to 5–8 Å). Greater cutoff values result in easier problems with comparatively higher number of

Input: The set S of atom pairs, for which the pairwise distances are available;
Output: The coordinate matrix X for determined atoms;
Find four atoms a_1, a_2, a_3, a_4 for which all pairwise distances are available and which do not lie on the same plane;
`// initialize the generation indices for these first four atoms`
$(G[a_1], G[a_2], G[a_3], G[a_4]) \leftarrow (0, 0, 0, 0)$;
Determine the coordinates $X[a_1]$, $X[a_2]$, $X[a_3]$, $X[a_4]$ using the pairwise distances between a_1, a_2, a_3, a_4;
repeat
  $N \leftarrow$ the set of all undetermined atoms, for each of which there are ≥ 4 adjacent atoms in S **and** these neighbours are determined **and** not on the same plane;
  $a \leftarrow \arg\min_{a \in N} \max_{\{a,b\} \in S, b \text{ is determined}} G[b]$;
  Determine $X[a]$ using approximate solution to a possibly overdetermined system of distance equations $\{d_{ab} = ||X[a] - X[b]|| : \{a, b\} \in S,\ b \text{ is determined}\}$, as described in Section 3.4;
  $G[a] \leftarrow 1 + \max_{\{a,b\} \in S, b \text{ is determined}} G[b]$;
until *no new atoms were determined* ;

Figure 5.1 Pseudo code for the Uniform Geometric Buildup Algorithm using Linear Least Squares Approximation (ULNLS).

available distances. If there is no cutoff, then all distances are available and the problem can be accurately solved using SVD, as described in Section 3.1. The number of all pairwise distances and the percentage of available distances after the cut off is reported for each of the benchmark problems in Table A.1.

Our goal is to evaluate the improvements in the quality of the MDGP solutions produced by algorithms that apply both design principles introduced in Chapter 4. First, we investigate the principle of *uniformness*: we compare the performance of the LNLS against the performance of the ULNLS algorithm. The two differ in the order of atom determination. In LNLS this order is not specified (in our implementation the order of atom indices in the MDGP is used). In ULNLS the order is specified using a greedy approach as the algorithm seeks to minimize the generation indices for newly-determined atoms. Next, we investigate the *decomposition* scheme: we compare the performance of ULNLS against ULNLSPD. The latter is simply the former wrapped into problem decomposition scheme (see Fig 4.3).

The performance of the LNLS algorithm was compared against ULNLS for two different implementations of LNLS and one implementation of ULNLS. The two sets of results for

Input: The set S of atom pairs, for which the distances are available;
Input: The limit L on the maximum generation index over atoms in each cluster;
Output: The list of clusters $\mathcal{C}$ of all determined atoms;
$\mathcal{C} \leftarrow$ empty list of atom clusters;
`// generate clusters`
while *can find four atoms* a_1, a_2, a_3, a_4, *for which all pairwise distances are available* ***and*** *these atoms are not on the same plane* ***and*** *at least one is not determined in each cluster in* $\mathcal{C}$ **do**
    $G \leftarrow$ empty list of atom generation indices;
    $C \leftarrow$ empty cluster;
    `// initialize generation indices for the first four determined atoms`
    $(G[a_1], G[a_2], G[a_3], G[a_4]) \leftarrow (0, 0, 0, 0)$;
    Determine the coordinates $C[a_1]$, $C[a_2]$, $C[a_3]$, $C[a_4]$ using the pairwise distances between a_1, a_2, a_3, a_4;
    repeat
        $N \leftarrow$ the set of all undetermined atoms, for each of which there are ≥ 4 adjacent atoms in S and these neighbours are determined and not on the same plane;
        $a \leftarrow \arg\min_{a \in N} \max_{\{a,b\} \in S, b \text{ is determined}} G[b]$;
        $C[a] \leftarrow$ linear least squares approximation for the solution of a possibly overdetermined system of distance equations
        $\{d_{ab} = ||X[a] - X[b]|| : \{a, b\} \in S,\ b \text{ is determined in } C\}$; `// see Section 3.4`
        $G[a] \leftarrow 1 + \max_{\{a,b\} \in S, b \text{ is determined in } C} G[b]$;
    until *no new atoms were determined* ;
end
`// merge clusters`
repeat
    foreach C_1, $C_2 \in \mathcal{C}$ *which overlap by* ≥ 4 *atoms and these atoms are not on the same plane* **do**
        Merge C_1 and C_2 using optimal translation and rotation of the set of overlapping atoms into a combined cluster C;
        Remove C_1 and C_2 from $\mathcal{C}$;
        Insert C into $\mathcal{C}$;
    end
until *no two clusters in* $\mathcal{C}$ *overlap by* ≥ 4 *atoms* ***and*** *these atoms are not on the same plane* ;

Figure 5.2 Uniform geometric buildup algorithm using linear least squares approximation, enhanced with the problem decomposition scheme (the ULNLSPD algorithm).

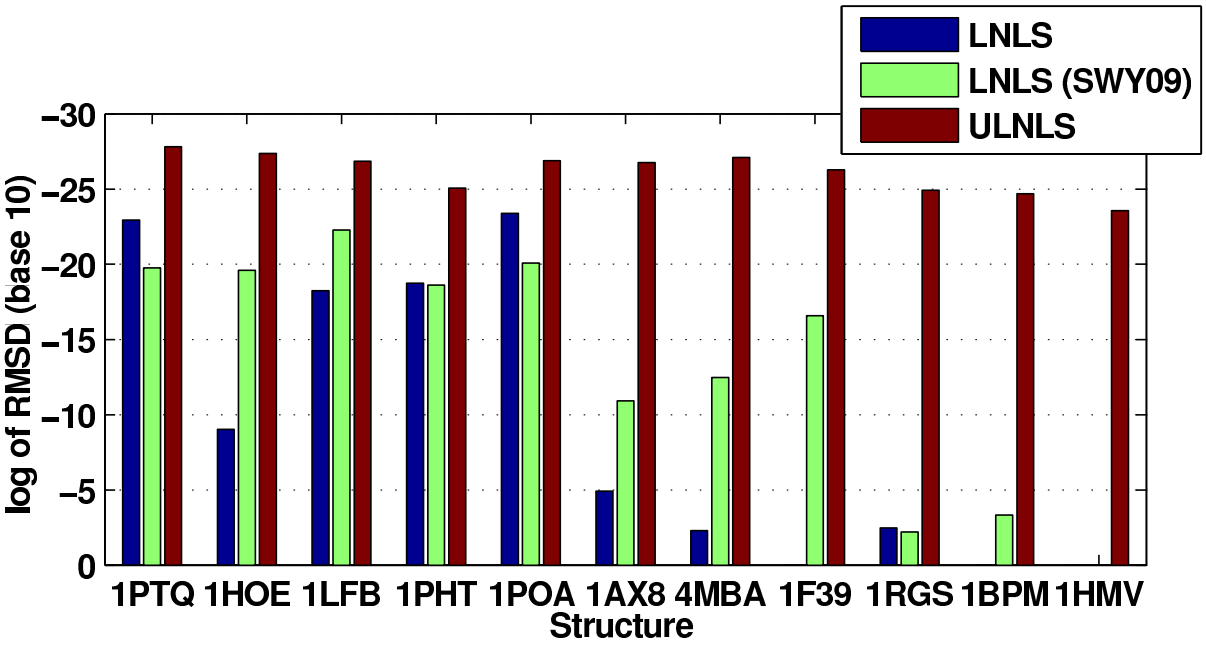

Figure 5.3 Performance of the 3 different geometric buildup algorithms: 1) our implementation of the LNLS GB algorithm (see Section 3.4); 2) the implementation of the LNLS GB algorithm from (SWY09); and 3) our implementation of the ULNLS algorithm. The performance was evaluated on a set of problems generated for the benchmark PDB databank structures which were used in (SWY09). The distance cutoff for MDGP problems generation was set to 6Å.

the LNLS algorithms were: 1) the results published in (SWY09), and 2) the results for an implementation of LNLS which was done as a part of this work. The ULNLS algorithm was proposed and implemented in this work (see Section 4). The benchmark MDGP instances from (SWY09) were used to produce numerical evaluations which allowed us to compare the performance of the three programs. The results, shown in Fig. 5.3, indicate that ULNLS outperforms LNLS. The only difference between the designs of the two algorithms is that the principle of uniformness is used in ULNLS. Thus, we conclude that using this principle leads to improved performance.

To see how the decomposition scheme affects the algorithm, we evaluated ULNLSPD on a number of benchmark MDGP instances, each of which was generated from a known protein structure. The numerical results for each individual instance are provided in Appendix A. The statistics for the distribution of base-10 logarithm of the RMSD, which is computed for each solution and its parent structure, are shown in Fig. 5.4. In general, solutions computed from

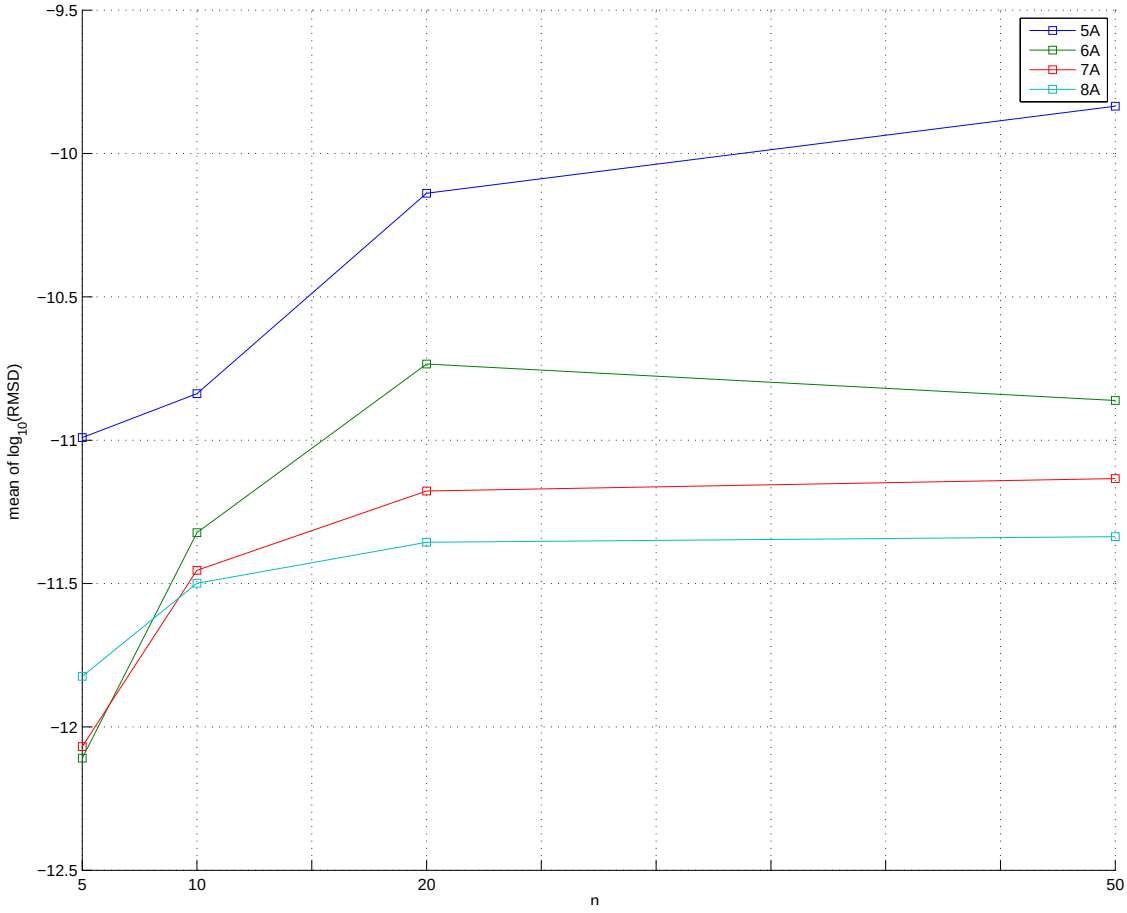

Figure 5.4 The mean of the log of RMSD for the ULNLSPD results at different cutoffs for different values of L. Only those structures that were solved for all 4 possible values of L were included in the computation of the mean. According to the data, more clusters (i.e., lower values of L) correlate well with better structures (i.e., lower values of the mean of the log of RMSD).

a larger number of small clusters, generated with lower values for L in ULNLSPD algorithm, are better than the ones generated with higher L.

To see how our algorithm compares against other published results for the solution of the MDGP, we collected results for solving the exact MDGP instances for a benchmark set of structures from (SWY09), (WW07), (GLS09), (BLTY05), see Fig. 5.5, 5.6. In these benchmarks, typically the best performing program is either our LNLS implementation or another geometric buildup implementation that uses non-linear least squares (SWY09). The global optimization approaches typically generate somewhat worse structures than recent geometric buildup imple-

mentations for this benchmark. It must be noted that for the larger molecules in the benchmark (1HMV, 1BPM), the ULNLS results are usually better, which suggests that the uniformness heuristic and the decomposition process may improve the scalability of the algorithm for larger structures. Furthermore, the NLLS algorithm does not use either the uniformness heuristic or the decomposition process, and therefore stands to benefit from either of these contributions.

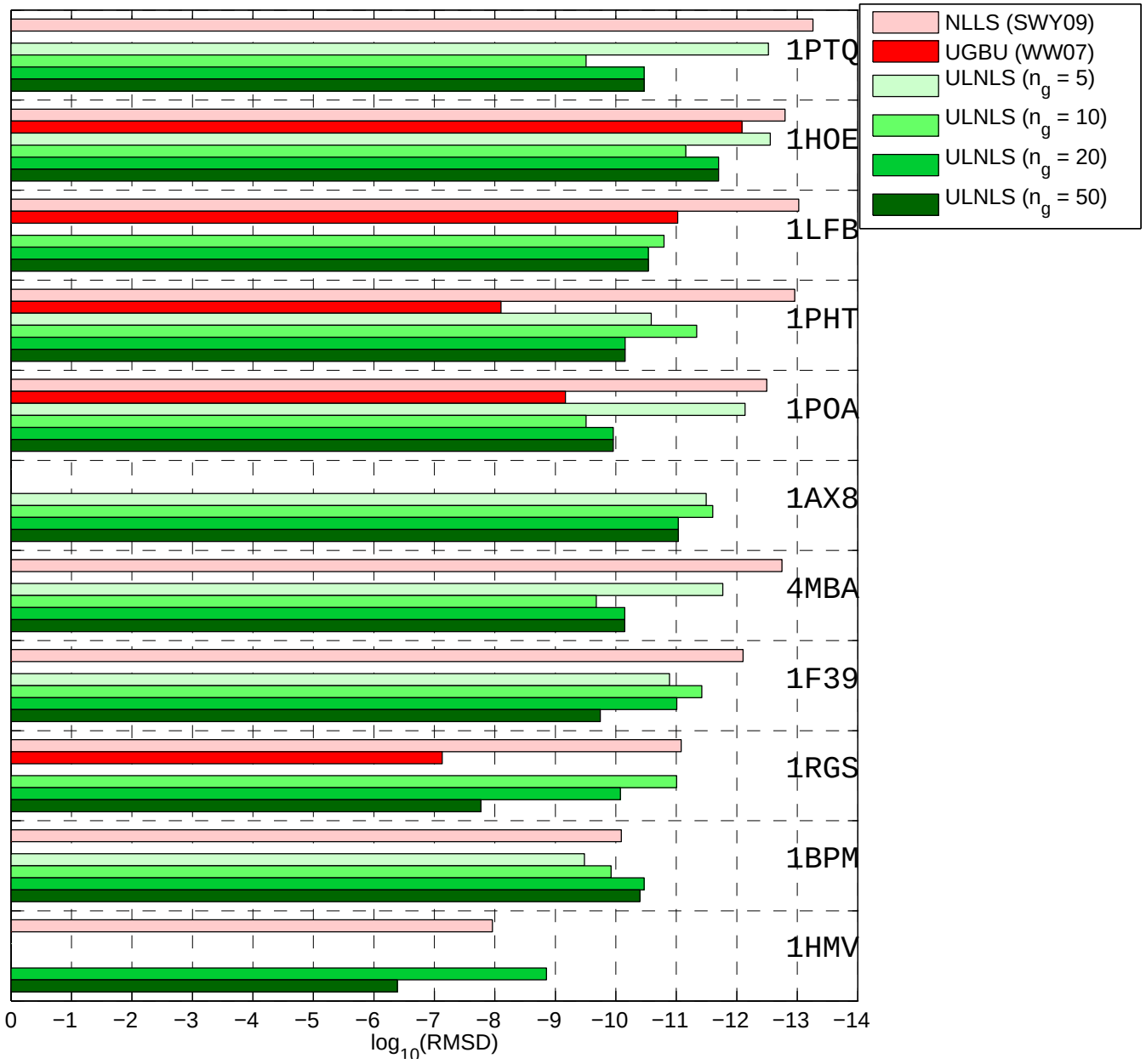

Figure 5.5 Comparison of the performance of our family of algorithms against previously published results on the benchmark problems with exact distances and the cutoff equal to 5Å. Only the results with the number of atoms in the output structure equal to the maximum such number over all surveyed algorithms are compared. The algorithms are: non-linear least squares geometric buildup (SWY09), updated geometric buildup (WW07) and the uniform linear least squares geometric buildup algorithm with decomposition from Figure 5.2, invoked with different values of n_g, i.e. 5, 10, 20, and 50.

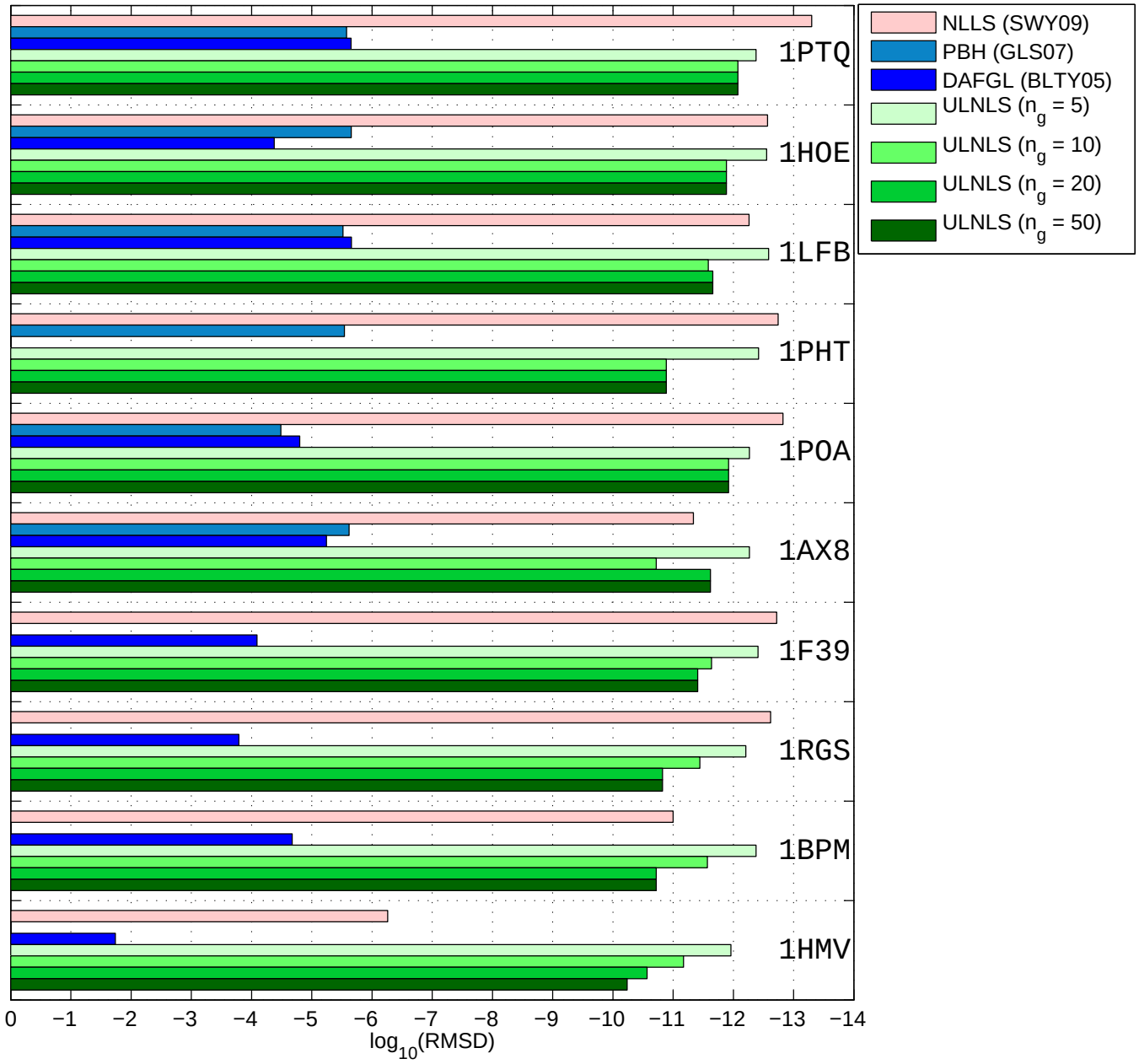

Figure 5.6 Comparison of the same algorithms on the benchmark problems with exact distances and the cutoff equal to 6Å. Only the results with the number atoms in the output structure equal to the size of the protein are compared (that is, if a given method failed or determined fewer atoms in the structure there is no bar). The algorithms are: non-linear least squares geometric buildup (SWY09), PBH global optimization algorithm (GLS09), DAFGL algorithm based on SDP relaxation (BLTY05) and the uniform linear least squares geometric buildup algorithm with decomposition from Figure 5.2, invoked with different values of n_g.

CHAPTER 6. Conclusions and Future Work

This work makes two contributions to the design of geometric buildup algorithms. The first contribution is a new procedure for the choice of the next atom to be determined that improves uniformness of the intermediate structures. The second contribution is a method for decomposing the MDGP into smaller subproblems that result in clusters that are merged together to form the solution of the whole MDGP. Both ideas are used to enhance a known geometric buildup algorithm using linear least squares approximation (LNLS). This results in a new uniform geometric buildup algorithm using linear least squares approximation with problem decomposition into clusters (CULNLS). This new algorithm was evaluated on a number of benchmark problems and was compared against LNLS to investigate how uniformness and decomposition affect the quality of the solutions. By analyzing the results on benchmark tests we can conclude that both uniformness and decomposition improve the quality of the solutions.

Even though the two proposed algorithmic enhancements were evaluated for a single geometric buildup algorithm, it is possible to apply them to any geometric buildup algorithm. Because the two principles that motivated the two contributions – i.e., the Occam's razor and the divide-and-conquer paradigm – are very general, it is reasonable to expect that any geometric buildup algorithm can be improved using the two proposed ideas.

There are several possible areas for future work. First, both uniformness and decomposition need to be incorporated into the otherwise best-performing geometric buildup algorithm with non-linear least squares approximation. Second, the procedure that selects the next atom to be determined that was introduced in this work to improve the uniformness is not necessarily the best such procedure. For example, it does not consider the error of the numerical solution to the overdetermined system of distance equations, which can be important for MDGP instances

with perturbed distance constraints or distance range constraints. Using statistically robust methods for the determination of the next atom's coordinates can also benefit such MDGP instances. Third, the procedure for merging the clusters that result from the decomposition process that is introduced in this work only uses the coordinates of those points that are shared between the clusters and ignores any distance constraints between the atoms that belong to distinct clusters. The merging procedure can also benefit from the use of statistically robust methods to improve the overall performance for MDGPs with perturbed distance constraints or distance range constraints.

Finally, in the geometric buildup framework, the number of the first atoms that are determined using pairwise distances does not necessarily have to be equal to four. There can be more starting atoms as long as all pairwise distances between them are known. To find these atoms, one can use polynomial approximations to the maximal subgraph problem to locate promising sets of the starting atoms that have all pairwise distance constraints available.

APPENDIX A. Benchmarks

The geometric buildup algorithm with cluster decomposition was evaluated on the benchmark set of protein structures with different cutoffs: Tables A.2, A.3, A.4, A.5.

id	n[i]	$\lvert\mathcal{D}_a\rvert$[ii]	cutoff 5Å		cutoff 6Å		cutoff 7Å		cutoff 8Å	
			$\lvert\mathcal{D}\rvert$[iii]	% of $\lvert\mathcal{D}_a\rvert$	$\lvert\mathcal{D}\rvert$	% of $\lvert\mathcal{D}_a\rvert$	$\lvert\mathcal{D}\rvert$	% of $\lvert\mathcal{D}_a\rvert$	$\lvert\mathcal{D}\rvert$	% of $\lvert\mathcal{D}_a\rvert$
1PTQ	402	80601	4399	5.46%	7088	8.79%	10302	12.78%	14023	17.40%
1HOE	558	155403	6299	4.05%	10178	6.55%	14936	9.61%	20423	13.14%
1LFB	641	6974	205120	3.40%	11435	5.57%	16602	8.09%	22519	10.98%
1PHT	814	330891	11033	3.33%	17695	5.35%	26299	7.95%	36077	10.90%
1POA	914	417241	10468	2.51%	16983	4.07%	24984	5.99%	34485	8.27%
1AX8	1003	502503	11542	2.23%	18795	3.74%	27286	5.43%	37130	7.39%
1GPV	1842	1695561	23863	1.41%	38006	2.24%	56949	3.36%	78876	4.65%
1RGS	2015	2029105	22784	1.12%	38020	1.87%	56298	2.77%	77513	3.82%
1BPM	3672	6739956	44789	0.66%	75152	1.11%	112940	1.68%	159303	2.37%
1HMV	7398	27361503	86288	0.32%	143196	0.52%	214498	0.78%	299939	1.10%
4MBA	1086	589155	12761	2.17%	20905	3.55%	30706	5.21%	42151	7.15%
1F39	1534	1175811	17300	1.47%	28532	2.43%	42678	3.63%	59551	5.06%
1HQQ	3944	7775596	46754	0.60%	77850	1.00%	118385	1.52%	167713	2.16%
1I7W	8629	37225506	106426	0.29%	176288	0.47%	261877	0.70%	367500	0.99%
1KDH	2846	4048435	33558	0.83%	55213	1.36%	82006	2.03%	114590	2.83%
1MQQ	5681	16134040	71698	0.44%	120357	0.75%	182452	1.13%	259691	1.61%
1NF7	6880	23663760	77080	0.33%	126806	0.54%	188868	0.80%	263951	1.12%
1NFB	5666	16048945	63380	0.39%	104380	0.65%	155342	0.97%	217373	1.35%
1QRB	4119	8481021	48091	0.57%	80382	0.95%	120947	1.43%	168240	1.98%
1RHJ	3740	6991930	45716	0.65%	76975	1.10%	115605	1.65%	163365	2.33%
1TJO	5459	14897611	64903	0.44%	107774	0.72%	161617	1.08%	226166	1.52%
1TOA	4292	9208486	51590	0.56%	86105	0.94%	129801	1.41%	183129	1.99%

[i] the number of atoms in the structure
[ii] the number of all distances, $\binom{n}{2}$
[iii] the number of distances remaining after the cutoff

Table A.1 MDGP sizes for benchmark structures at different cutoffs.

id	n[i]	$n_g = 5$			$n_g = 10$			$n_g = 20$			$n_g = 50$		
		$\|c_L\|$[ii]	n_c[iii]	RMSD[iv]	$\|c_L\|$	n_c	RMSD	$\|c_L\|$	n_c	RMSD	$\|c_L\|$	n_c	RMSD
1PTQ	402	402	11	3.0×10^{-13}	402	2	3.1×10^{-11}	402	1	3.4×10^{-11}	402	1	3.4×10^{-11}
1HOE	558	558	15	2.8×10^{-13}	558	3	7.0×10^{-12}	558	1	2.0×10^{-12}	558	1	2.0×10^{-12}
1LFB	641	640	16	4.0×10^{-13}	641	4	1.6×10^{-11}	641	1	2.9×10^{-11}	641	1	2.9×10^{-11}
1PHT	814	809	16	2.6×10^{-11}	809	3	4.6×10^{-12}	809	2	7.0×10^{-11}	809	2	7.0×10^{-11}
1POA	914	914	20	7.3×10^{-13}	914	3	3.1×10^{-10}	914	1	1.1×10^{-10}	914	1	1.1×10^{-10}
1AX8	1003	1003	24	3.2×10^{-12}	1003	7	2.5×10^{-12}	1003	1	9.3×10^{-12}	1003	1	9.3×10^{-12}
1GPV	1842	1842	34	9.9×10^{-12}	1842	10	4.5×10^{-12}	1842	2	1.3×10^{-9}	1842	1	1.3×10^{-8}
1RGS	2015	2007	56	4.8×10^{-12}	2010	17	9.9×10^{-12}	2010	8	8.4×10^{-11}	2010	6	1.7×10^{-8}
1BPM	3672	3669	84	3.3×10^{-10}	3669	16	1.2×10^{-10}	3669	4	3.4×10^{-11}	3669	3	4.0×10^{-11}
1HMV	7398	7372	210	1.4×10^{-10}	7388	47	1.1×10^{-10}	7389	14	1.4×10^{-9}	7389	9	4.1×10^{-7}
4MBA	1086	1083	29	1.7×10^{-12}	1083	7	2.1×10^{-10}	1083	2	7.1×10^{-11}	1083	2	7.1×10^{-11}
1F39	1534	1534	38	1.3×10^{-11}	1534	9	3.8×10^{-12}	1534	2	9.8×10^{-12}	1534	1	1.8×10^{-10}
1HQQ	3944	3926	95	5.5×10^{-11}	3925	22	2.0×10^{-9}	3938	7	5.9×10^{-8}	3938	2	3.8×10^{-8}
1I7W	8629	8616	234	3.2×10^{-10}	8623	46	2.6×10^{-9}	8624	9	1.3×10^{-9}	8624	4	5.7×10^{-7}
1KDH	2846	2842	68	6.4×10^{-11}	2846	12	2.8×10^{-10}	2846	2	1.6×10^{-10}	2846	1	3.3×10^{-9}
1MQQ	5681	5680	118	3.7×10^{-10}	5681	19	4.8×10^{-11}	5681	4	7.6×10^{-10}	5681	1	2.6×10^{-8}
1NF7	6880	6878	189	3.3×10^{-11}	6879	47	3.9×10^{-10}	6879	11	1.2×10^{-7}	6879	3	3.0×10^{-4}
1NFB	5666	5662	152	3.0×10^{-9}	5665	43	5.8×10^{-10}	5665	10	2.3×10^{-5}	5665	2	1.9×10^{-1}
1QRB	4119	4113	101	1.0×10^{-11}	4114	25	2.6×10^{-11}	4114	9	1.7×10^{-10}	4114	6	1.2×10^{-5}
1RHJ	3740	3740	74	7.0×10^{-8}	3740	11	4.6×10^{-12}	3740	3	2.0×10^{-8}	3740	1	1.6×10^{-9}
1TJO	5459	5459	146	6.7×10^{-11}	5459	29	1.5×10^{-10}	5459	7	4.4×10^{-11}	5459	1	8.0×10^{-10}
1TOA	4292	4279	106	3.7×10^{-12}	4280	24	2.43×10^{-11}	4280	8	8.0×10^{-9}	4280	6	2.1×10^{-8}

[i] the total number of atoms in the structure
[ii] the size of the largest cluster from the output of the algorithm.
[iii] the total number of clusters generated by the algorithm.
[iv] the RMSD is computed for the largest cluster c_L that is generated by the algorithm.

Table A.2 Results of geometric buildup with clusters algorithm described in Figure 5.2 parameterized by different number of generations applied to the benchmark set of protein structures. The value of the distance cutoff used to generate the problem instance is 5Å. The instances where the algorithm was able to determine the coordinates of all atoms with the RMSD less than 0.01 are indicated in green, i.e., the problem is solved.

id	n	$n_g = 5$			$n_g = 10$			$n_g = 20$			$n_g = 50$		
		$\|c_L\|$	n_c	RMSD	$\|c_L\|$	n_c	RMSD	$\|c_L\|$	n_c	RMSD	$\|c_L\|$	n_c	RMSD
1PTQ	402	402	4	4.2×10^{-13}	402	1	8.4×10^{-13}	402	1	8.4×10^{-13}	402	1	8.4×10^{-13}
1HOE	558	558	5	2.8×10^{-13}	558	1	1.3×10^{-12}	558	1	1.3×10^{-12}	558	1	1.3×10^{-12}
1LFB	641	641	6	2.6×10^{-13}	641	2	2.6×10^{-12}	641	1	2.2×10^{-12}	641	1	2.2×10^{-12}
1PHT	814	814	4	3.8×10^{-13}	814	1	1.3×10^{-11}	814	1	1.3×10^{-11}	814	1	1.3×10^{-11}
1POA	914	914	5	5.4×10^{-13}	914	1	1.2×10^{-12}	914	1	1.2×10^{-12}	914	1	1.2×10^{-12}
1AX8	1003	1003	9	5.4×10^{-13}	1003	2	1.9×10^{-11}	1003	1	2.4×10^{-12}	1003	1	2.4×10^{-12}
1GPV	1842	1842	15	2.6×10^{-12}	1842	4	3.0×10^{-11}	1842	1	5.4×10^{-11}	1842	1	5.4×10^{-11}
1RGS	2015	2015	18	6.2×10^{-13}	2015	4	3.6×10^{-12}	2015	1	1.5×10^{-11}	2015	1	1.5×10^{-11}
1BPM	3672	3672	25	4.2×10^{-13}	3672	3	2.7×10^{-12}	3672	1	1.9×10^{-11}	3672	1	1.9×10^{-11}
1HMV	7398	7398	59	1.1×10^{-12}	7398	12	6.7×10^{-12}	7398	2	2.7×10^{-11}	7398	1	5.8×10^{-11}
4MBA	1086	1086	7	2.4×10^{-13}	1086	2	4.2×10^{-12}	1086	1	1.7×10^{-12}	1086	1	1.7×10^{-12}
1F39	1534	1534	15	3.9×10^{-13}	1534	3	2.3×10^{-12}	1534	1	3.9×10^{-12}	1534	1	3.9×10^{-12}
1HQQ	3944	3943	30	1.3×10^{-11}	3944	7	1.2×10^{-11}	3944	1	2.8×10^{-10}	3944	1	2.8×10^{-10}
1I7W	8629	8629	62	1.9×10^{-12}	8629	11	3.5×10^{-12}	8629	4	2.6×10^{-11}	8629	1	1.7×10^{-10}
1KDH	2846	2846	24	5.3×10^{-13}	2846	5	4.8×10^{-12}	2846	1	1.7×10^{-11}	2846	1	1.7×10^{-11}
1MQQ	5681	5681	32	4.6×10^{-13}	5681	7	7.6×10^{-12}	5681	1	1.1×10^{-11}	5681	1	1.1×10^{-11}
1NF7	6880	6880	64	7.9×10^{-13}	6880	15	8.6×10^{-12}	6880	4	2.6×10^{-11}	6880	1	5.9×10^{-11}
1NFB	5666	5666	59	1.1×10^{-10}	5666	17	2.6×10^{-10}	5666	4	1.4×10^{-5}	5666	1	1.6×10^{-10}
1QRB	4119	4119	32	2.8×10^{-12}	4119	8	4.4×10^{-12}	4119	2	3.5×10^{-11}	4119	1	1.0×10^{-10}
1RHJ	3740	3740	23	4.6×10^{-13}	3740	6	3.3×10^{-12}	3740	1	7.2×10^{-12}	3740	1	7.2×10^{-12}
1TJO	5459	5459	36	4.0×10^{-13}	5459	11	3.3×10^{-12}	5459	3	2.5×10^{-11}	5459	1	6.9×10^{-11}
1TOA	4292	4292	28	7.3×10^{-13}	4292	6	3.1×10^{-12}	4292	1	1.8×10^{-11}	4292	1	1.8×10^{-11}

Table A.3 Results of geometric buildup with the clusters algorithm described in Figure 5.2 parameterized by different number of generations applied to the benchmark set of protein structures. The value of the distance cutoff used to generate the problem instance is 6Å. The instances where the algorithm was able to determine the coordinates of all atoms with the RMSD less than 0.01 are indicated in green, i.e., the problem is solved.

id	n	$n_g = 5$			$n_g = 10$			$n_g = 20$			$n_g = 50$		
		$\|c_L\|$	n_c	RMSD	$\|c_L\|$	n_c	RMSD	$\|c_L\|$	n_c	RMSD	$\|c_L\|$	n_c	RMSD
1PTQ	402	402	2	3.8×10^{-13}	402	1	6.8×10^{-13}	402	1	6.8×10^{-13}	402	1	6.8×10^{-13}
1HOE	558	558	3	7.3×10^{-13}	558	1	9.9×10^{-13}	558	1	9.9×10^{-13}	558	1	9.9×10^{-13}
1LFB	641	641	4	8.0×10^{-13}	641	1	1.1×10^{-12}	641	1	1.1×10^{-12}	641	1	1.1×10^{-12}
1PHT	814	814	2	8.8×10^{-13}	814	1	1.5×10^{-12}	814	1	1.5×10^{-12}	814	1	1.5×10^{-12}
1POA	914	914	2	4.7×10^{-13}	914	1	7.6×10^{-13}	914	1	7.6×10^{-13}	914	1	7.6×10^{-13}
1AX8	1003	1003	5	9.6×10^{-13}	1003	1	2.3×10^{-12}	1003	1	2.3×10^{-12}	1003	1	2.3×10^{-12}
1GPV	1842	1842	8	9.0×10^{-13}	1842	1	1.9×10^{-12}	1842	1	1.9×10^{-12}	1842	1	1.9×10^{-12}
1RGS	2015	2015	10	9.0×10^{-13}	2015	2	5.6×10^{-12}	2015	1	1.4×10^{-11}	2015	1	1.4×10^{-11}
1BPM	3672	3672	6	6.3×10^{-13}	3672	5	5.2×10^{-12}	3672	1	1.3×10^{-11}	3672	1	1.3×10^{-11}
1HMV	7398	7398	29	9.5×10^{-13}	7398	5	9.5×10^{-12}	7398	2	5.3×10^{-11}	7398	1	3.0×10^{-11}
4MBA	1086	1086	4	5.2×10^{-13}	1086	1	1.4×10^{-12}	1086	1	1.4×10^{-12}	1086	1	1.4×10^{-12}
1F39	1534	1534	7	7.3×10^{-13}	1534	2	3.6×10^{-12}	1534	1	3.0×10^{-12}	1534	1	3.0×10^{-12}
1HQQ	3944	3944	15	7.2×10^{-13}	3944	3	5.1×10^{-12}	3944	1	9.2×10^{-12}	3944	1	9.2×10^{-12}
1I7W	8629	8629	27	1.6×10^{-12}	8629	6	9.3×10^{-12}	8629	2	3.6×10^{-11}	8629	1	6.7×10^{-11}
1KDH	2846	2846	8	6.8×10^{-13}	2846	2	3.3×10^{-12}	2846	1	9.5×10^{-12}	2846	1	9.5×10^{-12}
1MQQ	5681	5681	15	7.5×10^{-13}	5681	4	5.9×10^{-12}	5681	1	8.5×10^{-12}	5681	1	8.5×10^{-12}
1NF7	6880	6880	36	1.8×10^{-12}	6880	7	1.0×10^{-11}	6880	2	7.1×10^{-11}	6880	1	2.1×10^{-10}
1NFB	5666	5666	29	9.9×10^{-13}	5666	6	1.7×10^{-11}	5666	1	3.9×10^{-11}	5666	1	3.9×10^{-11}
1QRB	4119	4119	16	8.6×10^{-13}	4119	4	5.7×10^{-12}	4119	2	3.2×10^{-11}	4119	1	9.9×10^{-11}
1RHJ	3740	3740	12	2.4×10^{-12}	3740	3	6.1×10^{-12}	3740	1	5.9×10^{-12}	3740	1	5.9×10^{-12}
1TJO	5459	5459	25	8.8×10^{-13}	5459	5	6.4×10^{-12}	5459	2	5.0×10^{-11}	5459	1	4.7×10^{-11}
1TOA	4292	4292	17	1.0×10^{-12}	4292	3	5.8×10^{-12}	4292	1	1.2×10^{-11}	4292	1	1.2×10^{-11}

Table A.4 Results of geometric buildup with the clusters algorithm described in Figure 5.2 parameterized by different number of generations applied to the benchmark set of protein structures. The value of the distance cutoff used to generate the problem instance is 7Å. The instances where the algorithm was able to determine the coordinates of all atoms with the RMSD less than 0.01 are indicated in green, i.e. the problem is solved.

id	n	$n_g = 5$			$n_g = 10$			$n_g = 20$			$n_g = 50$		
		$\|c_L\|$	n_c	RMSD	$\|c_L\|$	n_c	RMSD	$\|c_L\|$	n_c	RMSD	$\|c_L\|$	n_c	RMSD
1PTQ	402	402	2	1.3×10^{-12}	402	1	4.8×10^{-13}	402	1	4.8×10^{-13}	402	1	4.8×10^{-13}
1HOE	558	558	2	7.4×10^{-13}	558	1	7.6×10^{-13}	558	1	7.6×10^{-13}	558	1	7.6×10^{-13}
1LFB	641	641	3	1.0×10^{-12}	641	1	8.8×10^{-13}	641	1	8.8×10^{-13}	641	1	8.8×10^{-13}
1PHT	814	814	2	8.5×10^{-13}	814	1	8.8×10^{-13}	814	1	8.8×10^{-13}	814	1	8.8×10^{-13}
1POA	914	914	2	5.5×10^{-13}	914	1	7.4×10^{-13}	914	1	7.4×10^{-13}	914	1	7.4×10^{-13}
1AX8	1003	1003	2	8.8×10^{-13}	1003	1	1.1×10^{-12}	1003	1	1.1×10^{-12}	1003	1	1.1×10^{-12}
1GPV	1842	1842	4	1.2×10^{-12}	1842	1	1.1×10^{-12}	1842	1	1.1×10^{-12}	1842	1	1.1×10^{-12}
1RGS	2015	2015	4	1.1×10^{-12}	2015	2	9.4×10^{-12}	2015	1	1.2×10^{-11}	2015	1	1.2×10^{-11}
1BPM	3672	3672	7	9.6×10^{-13}	3672	2	7.2×10^{-12}	3672	1	9.9×10^{-12}	3672	1	9.9×10^{-12}
1HMV	7398	7398	14	1.2×10^{-12}	7398	3	8.1×10^{-12}	7398	1	2.9×10^{-11}	7398	1	2.9×10^{-11}
4MBA	1086	1086	3	4.6×10^{-11}	1086	1	9.6×10^{-13}	1086	1	9.6×10^{-13}	1086	1	9.6×10^{-13}
1F39	1534	1534	5	1.3×10^{-12}	1534	1	3.6×10^{-12}	1534	1	3.6×10^{-12}	1534	1	3.6×10^{-12}
1HQQ	3944	3944	8	1.0×10^{-12}	3944	2	9.6×10^{-12}	3944	1	7.9×10^{-12}	3944	1	7.9×10^{-12}
1I7W	8629	8629	16	2.7×10^{-12}	8629	3	1.1×10^{-11}	8629	2	4.0×10^{-11}	8629	1	4.0×10^{-11}
1KDH	2846	2846	6	1.0×10^{-12}	2846	2	8.2×10^{-12}	2846	1	8.0×10^{-12}	2846	1	8.0×10^{-12}
1MQQ	5681	5681	10	5.0×10^{-12}	5681	1	7.6×10^{-12}	5681	1	7.6×10^{-12}	5681	1	7.6×10^{-12}
1NF7	6880	6880	18	2.8×10^{-12}	6880	4	8.6×10^{-12}	6880	1	2.0×10^{-11}	6880	1	2.0×10^{-11}
1NFB	5666	5666	12	2.6×10^{-12}	5666	3	6.8×10^{-12}	5666	1	1.6×10^{-11}	5666	1	1.6×10^{-11}
1QRB	4119	4119	12	1.2×10^{-12}	4119	4	9.4×10^{-12}	4119	2	4.8×10^{-11}	4119	1	1.3×10^{-10}
1RHJ	3740	3740	7	1.5×10^{-12}	3740	1	5.3×10^{-12}	3740	1	5.3×10^{-12}	3740	1	5.3×10^{-12}
1TJO	5429	5459	14	1.2×10^{-12}	5459	4	1.1×10^{-11}	5459	1	3.7×10^{-11}	5459	1	3.7×10^{-11}
1TOA	4292	4292	8	9.2×10^{-13}	4292	3	8.8×10^{-12}	4292	1	9.1×10^{-12}	4292	1	9.1×10^{-12}

Table A.5 Results of geometric buildup with the clusters algorithm described in Figure 5.2 parameterized by different number of generations applied to the benchmark set of protein structures. The value of the distance cutoff used to generate the problem instance is 8Å. The instances where the algorithm was able to determine the coordinates of all atoms with the RMSD less than 0.01 are indicated in green, i.e. the problem is solved.

BIBLIOGRAPHY

[Y38] Young, G. and A. Householder. Discussion of a set of points in terms of their mutual distances. *Psychometrika 3(1). pp. 19–22, Springer, 1938.*

[B70] Blumenthal, L. Theory and applications of distance geometry. *Oxford University Press, 1953*

[T58] Torgeson, W.S. Theory and Method of Scaling. *Wiley, New York, 1958*

[S79] James B. Saxe. Embeddability of Weighted Graphs in k-space is strongly NP-hard. *Proceedings of the 17th Allerton Conference on Communication, Control, and Computing, 1979*

[SS85] Sippl, M. and H. Scheraga (1985). Solution of the embedding problem and decomposition of symmetric matrices. *Proceedings of the National Academy of Sciences 82(8): 2197.*

[SS86] Sippl, M. and H. Scheraga (1986). Cayley-Menger coordinates. *Proceedings of the National Academy of Sciences 83(8): 2283.*

[GH88] G.M. Crippen and T. Havel. Distance Geometry and Molecular Conformation. *Wiley, New York, 1988.*

[H92] Bruce Hendrickson. Conditions for unique graph realizations. *SIAM J. Comput. 21 (1):65-84, February 1992.*

[H95] Bruce Hendrickson. The molecule problem: exploiting structure in global optimization. *SIAM J. Optim., 5, 1995, 835-857.*

[MW95] Jorge J. Moré and Zhijun 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.

[ZBS97] Zhihong Zou, Richard H. Bird, Robert B. Schnabel. A Stochastic/Perturbation Global Optimization Algorithm for Distance Geometry Problems. *Journal of Global Optimization. 11(1) 91-105.*

[WWY07] Di Wu and Zhijun Wu and Yaxiang Yuan. The solution of the distance geometry problem in protein modeling via geometric buildup. *International Symposium on Mathematical and Computational Biology, Rubem Mondaini, ed., 2007.*

[GL89] G.H. Golub and C.F. van Loan. Matrix Computations. *Johns Hopkins University Press, 1989.*

[DW02] Qunfeng Dong and Zhijun Wu. A linear-time algorithm for solving the molecular distance geometry problem with exact inter-atomic distances. *Journal of Global Optimization, vol. 200, pp. 365-375, 2002.*

[DW03] Qunfeng Dong and Zhijun Wu. A geometric build-up algorithm for solving the molecular distance geometry problem with sparse distance data. *Journal of Global Optimization, vol. 26, pp. 321-333, 2003.*

[WW07] Di Wu and Zhijun Wu. An updated geometric build-up algorithm for solving the molecular distance geometry problem with sparse distance data. *Journal of Global Optimization, vol. 37 (4), pp. 661-673, 2007.*

[SWY09] Atilla Sit and Zhijun Wu and Yaxiang Yuan. A geometric buildup algorithm for the solution of the distance geometry problem using least-squares approximation. *Bulletin of Mathematical Biology, vol. 71 (8), pp.1914-1933, 2009.*

[GLS09] Andrea Grosso and Marco Locatelli and Fabio Schoen. Solving molecular distance geometry problems by global optimization algorithms. *Journal of Computational Optimization and Applications, vol. 43 (1), pp. 23-37, 2009.*

[BLTY05] Pratik Biswas and Tzu-Chen Liang and Kim-Chuan Toh and Yinyu Ye. An SDP based approach for anchor-free 3D graph realization. *Technical Report, Operations Research, Stanford University, Stanford, CA, 2005.*

[BTY08] Pratik Biswas and Kim-Chuan Toh and Yinyu Ye. A Distributed SDP Approach for Large-Scale Noisy Anchor-Free Graph Realization with Applications to Molecular Conformation. *Siam Journal on Scientific Computing 30 (3), pp. 1251-1277 (2008).*

[BHN03] H.M. Berman and K. Henrick and H. Nakamura. Announcing the worldwide Protein Data Bank. *Nature Structural Biology 10 (12), p. 980 (2003).*

[MC94] V.N. Maiorov and G.M. Crippen. Significance of root-mean-square deviation in comparing three-dimensional structures of globular proteins. *Journal of Molecular Biology 235(2), pp. 625-34 (1994).*

[Sto01] D. Stork. Foundations of Occam's razor and parsimony in learning. *NIPS 2001 Workshop. http://www.rii.ricoh.com/~stork/OccamWorkshop.html*

[Knuth98] D. Knuth. The Art of Computer Programming: Volume 3, Sorting and Searching, 2nd ed. *Addison-Wesley, 1998.*

[T98] M. Trosset. Applications of Multidimensional Scaling to Molecular Conformation. *Mining and Modeling Massive Data Sets in Science, Engineering, and Business with a Subtheme in Environmental Statistics 29(1) pp. 148-152 (1998)*

[GHR93] W. Glunt, T.L. Hayden, and M. Raydan. Molecular conformations from distance matrices. *Journal of Computational Chemistry 14, pp. 114-120, (1993)*

[K76] W. Kabsch. A solution of the best rotation to relate two sets of vectors. *Acta Crystallographica A32 pp. 922-923 (1976).*