Abstract—The Distance Geometry Problem (DGP) is defined as the problem of finding the spatial representation of a set of points, given the distances between them. An existing method in the literature that propose to solve this problem is the Stochastic Proximity Embedding (SPE), a simple, robust and self-organizing method that starts with an initial random configuration, selects pairs of points at random and adjust their distances based on a learning rate. The purpose is generate results that are close to the given set of relations between the objects. Therefore, this method is not fully effective to deal with uncertain distances, as expected in real world situation.

The Molecular Distance Geometry Problem (MDGP) arises from the protein structure determination problem, and consists in finding Cartesian coordinates of the atoms in a molecule, based on the set of some interval distances obtained by Nuclear Magnetic Resonance (NMR). In this work, we propose a method to address the MDGP based on SPE. To determine one protein structure, a small subset of atoms is selected and optimized by an interval distance version of SPE. Iteratively, this subset is increased until the full protein is determined. We applied this method on artificially created NMR data and obtained significant results.

Keywords: Protein Structure, NMR, Distance Geometry

1. Introduction

The Distance Geometry Problem (DGP) arises on the need to determine the coordinates for a set of objects geographically distributed using an incomplete and imprecise set of distances. Given an integer $K > 0$, the problem is embedding a simple undirected graph $G = (V, E, r)$, whose edges are weighted by a nonnegative function $r : E \rightarrow \mathbb{R}_+$. Thus, it is necessary to find a realization $x : V \rightarrow \mathbb{R}^K$ such as the Euclidean distance $d$, among the pair of points $\{i, j\}$, be equal to the edges weight:

$$\forall\{i, j\} \in E, \|x(i) - x(j)\| = r(\{i, j\})$$

(1)

Through this article we will adopt $\{x_i, x_i, r_{ij}\}$ instead of $\{x(i), x(j), r(\{i, j\})\}$.

The graph embedding problem is $NP$-Complete in linear case and $NP$-Hard for $K > 1$ [19], although this problem can be addressed in linear time for a sufficient dense graph [5].

The problem is is present in several notable areas, like sensor network localization [9], architecture [11], [8] and protein structure determination [10], [7], [14], this work focus on protein structure determination.

The word protein derives from the Greek protos, meaning “primary”, “most important” or “standing in front” [18]. These macromolecules are composed of a long polypeptide chain, molding a complex and stable structure. The protein conformation is essential to determine its functionality and they are used in almost all essentials biological processes, e.g., transporting oxygen from the lungs to other organs and tissues in all vertebrates. Therefore, knowledge about the protein structure is essential to analyze and manipulate it, considering all the possible interactions between molecules.

Currently, there are two major methods to determine protein structures: X-ray crystallography and Nuclear Magnetic Resonance (NMR). The x-ray crystallography works with cristalized proteins; using x-ray beam to determine the density of electrons and after that, the protein structure. The protein crystallization process imposes some restrictions not present in NMR method.

Using NMR method, the protein is submitted to an external magnetic field which induces the alignment of atoms spin in the observed nuclei. The interference on those spins can be measured hence their distance.

The NMR measurements are not precise and the resulting distance is a interval value, typically one angstrom wide, of some atoms. Along with NMR imprecise distances, the distances of atoms separated by one or two covalent bonds can be determined precisely, once the chemical composition of the protein is known and the covalent bonds and angles are stable.

Fortunately, proteins chemical composition are known, consequently providing known distances. For example, if two atoms are chemically bonded or bonded to a common atom, it is possible to determine their relative distance. That distance is not precise, but can be considered fixed and the small variations can be used in NMR experiments.
Thus, this method provides an interval weighted graph, which represents the backbone with a sparse set of its uncertain distances. The final possible protein conformation is determined by solving a molecular distance geometry problem.

The Molecular Distance Geometry Problem (MDGP) is a variation of DGP with some interval distances

\[ l_{ij} \leq r_{ij} \leq u_{ij} \tag{2} \]

where \( l_{ij} \) and \( u_{ij} \) are the lower and upper bounds for the distance from atom \( i \) do \( j \). Therefore each realization \( x : V \rightarrow \mathbb{R}^3 \) is

\[ \forall \{i, j\} \in E, l_{ij} \leq \|x(i) - x(j)\| \leq u_{ij} \tag{3} \]

where each atom is associated with one vertex and the known distances - accurate or interval - to one edge connecting the respective vertices. Like DGP, MDGP can be solved in linear time given enough precise distances, but using only NMR imprecise distances and covalent bonds precise distances and considering the experimental errors [6], as suggested in [4] the solution for MDGP can be formulated as a global optimization problem:

\[
\min_{x \in \mathbb{R}^{Kn}} \sum_{\{i, j\} \in E} \min^2\left(\left\|x_i - x_j\right\|^2 - l_{ij}, 0\right) + \max^2\left(\left\|x_i - x_j\right\|^2 - u_{ij}, 0\right) \tag{4}
\]

This problem can also be called of Interval Molecular Distance Geometry Problem (iMDGP) [16]. The following are some methods to solve the MDGP.

The Geometry Build-Up algorithm (GBU) [5] uses four non-planar atoms, called geometric base, and calculate their coordinates. Then, it proceeds iteratively solving linear systems and determining the position of the remaining atoms in the molecule, given the distances between the base atoms and the atom to be determined. In [20] was proposed a new version of GBU, that can minimize the effects of errors caused by floating point operations.

The DGSOL method [17] aims to show that continuation algorithms, based on Gaussian smoothing, can be used to develop an efficient code for the solution of DGP. The algorithm searches for a global minimizer of the function based on the Gauss-Hermite transform. The problem of DGSOL is the dependence of structures on the data distance.

Branch-and-Prune (B&P) is another iterative method to address the MDGP [15]. It uses 3 already positioned atoms and three known distances to determine the position of the next atom until the entire protein is determined. As three atoms are not sufficient to uniquely determine to position of one point in \( \mathbb{R}^3 \), each calculated atom can assume two distinct positions. At each step a new search branch is created, so the size of the solution can increase exponentially.

The pruning is performed after each step, another available distances are used to cut the branches:

1) both positions are possible: the both branches are explored;
2) only one branch is possible: the possible branch is stored and the other is pruned;
3) neither position is possible: the both branches are pruned and the search is backtracked.

The idea of B&P is use only exact distances - from molecular geometry - to position the atoms and use the intervalar distances - from NMR - in the pruning process.

An overview of the presented methods can be found in [16]. In this article we will use the Stochastic Proximity Embedding (SPE) as base for the development of new methods.

### 1.1 Stochastic Proximity Embedding

Stochastic Proximity Embedding (SPE), introduced in [2], [1], creates one embedding trough a continuous optimization. Given \( n \) objects and a set of expected distances \( r \), the method starts with an initial random configuration and iteratively refines it by repeatedly selecting two points \( \{u, v\} \) at random. The distance \( d_{uv} \) is calculated and the their coordinates are updated as follow:

\[
x_i \leftarrow x_i + \lambda \frac{r_{ij} - d_{ij}}{4d_{ij}} (x_j - x_i) \tag{5}
\]

\[
x_j \leftarrow x_j + \lambda \frac{r_{ij} - d_{ij}}{4d_{ij}} (x_i - x_j) \tag{6}
\]

The \( \lambda \) is the learning rate, used to avoid oscillations, it starts with 1 and decreases until the value becomes close enough to 0. This process is described in Algorithm 1.

**Data:** protein graph, \( \lambda, \lambda_\Delta, C \)

**Result:** protein structure

1. while \( \lambda \geq 0 \) do
2.  for \( (i = 0; i \leq C; i = i + 1) \) do
3.     random selection of \( u \) and \( v \);
4.     update \( u \) and \( v \);
5.  end
6. \( \lambda \leftarrow \lambda - \lambda_\Delta \);
7. end

**Algorithm 1:** SPE method

This article differs from others existing in the literature because we present solutions to the MDGP with uncertain data. The rest of the paper is organized as follows. Section 2 describes the proposed methods, Section 3 presents the computational experiments performed and their results. In Section 4 the conclusions are presented.
2. Proposed Methods

Our method consists in a structural determination based on interval inter-atomic distances. That process is very important when we consider that in real world, the data obtained in NMR are ruled by minimum and maximum constraints: the distance relations, before exacts, now have minimum ($r_{ij}^{\text{min}}$) and maximum ($r_{ij}^{\text{max}}$) values.

2.1 Stochastic Proximity Embedding Interval

The Stochastic Proximity Embedding Interval ($SPE_i$) is similar to the original $SPE$ [2], [1]; but the update function is modified to work with intervalar constraints ($r_{ij}^{\text{min}}, r_{ij}^{\text{max}}$):

If $d_{ij} < r_{ij}^{\text{min}}$:

\[
x_i \leftarrow x_i + \frac{r_{ij}^{\text{min}} - d_{ij}}{4d_{ij}}(x_i - x_j)
\]

(7)

\[
x_j \leftarrow x_j + \frac{r_{ij}^{\text{min}} - d_{ij}}{4d_{ij}}(x_j - x_i)
\]

(8)

If $d_{ij} > r_{ij}^{\text{max}}$:

\[
x_i \leftarrow x_i + \frac{r_{ij}^{\text{max}} - d_{ij}}{4d_{ij}}(x_i - x_j)
\]

(9)

\[
x_j \leftarrow x_j + \frac{r_{ij}^{\text{max}} - d_{ij}}{4d_{ij}}(x_j - x_i)
\]

(10)

The remaining of the method is the same as presented in [2], [1].

2.2 Progressive SPE

In this section we introduce a progressive version of SPE ($SPE_P$). To develop this variation we considered:

- The atoms of an protein are ordered, following the backbone. We use this fact and restrict the selection of $i$ and $j$ to the first $S$ atom of the protein. The value of $S$ is increased until its achieves the size of protein;
- Using the ordering, we introduce the notion of older and newer atoms. When we update the $i$ and $j$ atoms, only the newer atom is changed.

The update function, reformulated:

If $d_{ij} < r_{ij}^{\text{min}}$:

\[
x(j) \leftarrow x(j) + \frac{r_{ij}^{\text{min}} - d_{ij}}{2d_{ij}}(x_j - x_i)
\]

(11)

If $d_{ij} > r_{ij}^{\text{max}}$:

\[
x(j) \leftarrow x(j) + \frac{r_{ij}^{\text{max}} - d_{ij}}{2d_{ij}}(x_j - x_i)
\]

(12)

The original algorithm was also modified and presented in Algorithm 2.

Data: protein graph, $\lambda, \lambda_\Delta, C, p$;
Result: protein structure

1. while $S \leq T$ do
2. \hspace{1cm} while $\lambda \geq 0$ do
3. \hspace{2cm} for $(i \leftarrow 0; i \leq C; i++)$ do
4. \hspace{3cm} random selection of $i$;
5. \hspace{3cm} random selection of $j \neq i$;
6. \hspace{3cm} if $i > j$ then
7. \hspace{4cm} $(i, i) \leftarrow (j, i)$
8. \hspace{3cm} end
9. \hspace{2cm} update $i$ and $j$;
10. \hspace{1cm} end
11. \hspace{1cm} $\lambda \leftarrow \lambda - \lambda_\Delta$;
12. \hspace{1cm} end
13. \hspace{1cm} $S \leftarrow S + p$;
14. end

Algorithm 2: Progressive Method

2.3 Sliding Window

The last method ($SPE_{SW}$) is based on sliding windows: the range for sampling is limited in its minimum and maximum, a parameter $p$ is defined to determine the variation of the sliding window (see Algorithm 3), the first atom chosen is sampled from $S/2$ to $S$ range - using the backbone ordering; the second one is any neighbor of the first. The update function is the same used in $SPE_P$.

Data: protein graph, $\lambda, \lambda_\Delta, C, p$;
Result: protein structure

1. $S \leftarrow p$;
2. while $S \leq T$ do
3. \hspace{1cm} while $\lambda \geq 0$ do
4. \hspace{2cm} for $(k \leftarrow 0; k \leq C; k++)$ do
5. \hspace{3cm} $i \leftarrow$ random point of $\left[\frac{S}{2}, S\right]$;
6. \hspace{3cm} $j \leftarrow$ random neighbor of $i$;
7. \hspace{3cm} update $i$ and $j$;
8. \hspace{2cm} end
9. \hspace{1cm} $\lambda \leftarrow \lambda - \lambda_\Delta$;
10. \hspace{1cm} end
11. \hspace{1cm} $S \leftarrow S + p$;
12. end

Algorithm 3: Sliding Window method

This method can efficiently solve problems with sparse and inexact data, providing satisfactory results (Section 3).

3. Computational Experiments

To perform the experiments, we used artificial backbones [12] and simulated NMR experiments using this rules: with atoms separated up to 5 Å, we use a interval distance between 2 to 3 Å, 3 to 4 Å and 4 to 5 Å, according to the observed real distance. Also, exact distances for atoms separated up to 2 covalent bonds [1].
Table 1: Experimental results obtained from $SPE_I$, based on different initial configuration.

<table>
<thead>
<tr>
<th>$T^1$</th>
<th>$p^2$</th>
<th>RMSD $^3$</th>
<th>Restric $^4$(%)</th>
<th>LDE $^5$</th>
<th>Time(s)$^6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td></td>
<td>1.860</td>
<td>0.519</td>
<td>0.012</td>
<td>300</td>
</tr>
<tr>
<td>50</td>
<td></td>
<td>2.772</td>
<td>0.676</td>
<td>0.002</td>
<td>120</td>
</tr>
<tr>
<td>100</td>
<td></td>
<td>5.552</td>
<td>0.582</td>
<td>0.016</td>
<td>180</td>
</tr>
<tr>
<td>250</td>
<td></td>
<td>16.835</td>
<td>0.471</td>
<td>0.030</td>
<td>240</td>
</tr>
</tbody>
</table>

1: protein size 2: number of cycles 3: final value of RMSD 4: final value of Restric $\times 100$ 5: final value of ScoreLDE 6: total time of the process

To evaluate the results, we applied the following metrics:

- **Root-mean-square Deviation (RMSD)** The RMSD is used to measure the similarity of two proteins, hence is commonly used to quantify the quality of a generated protein given a correct one[3]. The RMSD of two structures is computed as:

$$ \frac{1}{n} \sum_{i=1}^{n} \|x_i - y_i\| $$

with $n$ the size of the protein, $\{x_i\}$ is the original protein - the artificial backbone - and $\{y_i\}$ is the generated protein rotated and translated to minimize the resulting RMSD;

- **Satisfied constraints percentage** Given one protein is satisfied. For interval distances the distance is calculated and verified if its lays on the interval; for exact ones we considered a tolerance of 0.1 Å;

- **Largest Distance Error (LDE)** LDE is one of the most used penalty function for the MDGP [13] and is defined as

$$ \sum_{(i,j) \in E} \min\left(\frac{||x_i - x_j|| - l_{ij}}{l_{ij}}, 0\right) + \max\left(\frac{||x_i - x_j|| - u_{ij}}{u_{ij}}, 0\right) $$

(14)

All the algorithms were implemented in Python and the experiments performed on Intel Core i3 of 2nd generation, with 4GB RAM running Linux Ubuntu version 12.04.

The tests were performed with $\lambda$ starting at 1.0 and $\Delta$ at 0.001, these values are obtained empirically. The results for $SPE_I$ are presented in Table 1 and one result is showed in Figure 1.

The results for $SPE_P$ and $SPE_SW$ are in Table 2 and Table 3, respectively. In Figure 2 we show one result with 25 atoms for $SPE_P$. In Figure 3 we can see better results to the application of PM in the same sample size of Tab. 1, that can be confirmed comparing Fig. 1 and Fig. 2.

### 4. Conclusions

We presented three different methods, based on SPE, to solve MDGP with interval distances. The regular SPE method with intervalar distances showed non-scalable. The $SPE_{SW}$ and $SPE_P$ showed up more robust and able to solve successfully the problem for proteins of up to 250 atoms.

### Acknowledgments

The authors would like to thank the Brazilian National Council for Scientific and Technological Development (CNPq), Research Support Foundation of Goiás State (FAPEG) and Federal University of Uberlândia (PROPP grant 04/1014/074).

### References


Fig. 2: In red the original protein with 25 atoms and in green the protein produced by $SPE_P$ with $C = 6,000$.

Table 3: Final results of Sliding Window method with different numbers of cycles and $p = 2i + 10$. The method was also applied to a protein with 300 atoms.

<table>
<thead>
<tr>
<th>$T$</th>
<th>$C$</th>
<th>RMSD</th>
<th>Restrict ($%$)</th>
<th>ScoreLDE</th>
<th>Time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>6000</td>
<td>1.310</td>
<td>0.357</td>
<td>0.054</td>
<td>600</td>
</tr>
<tr>
<td>50</td>
<td>6000</td>
<td>5.516</td>
<td>0.519</td>
<td>0.041</td>
<td>1800</td>
</tr>
<tr>
<td>100</td>
<td>6000</td>
<td>4.990</td>
<td>0.510</td>
<td>0.034</td>
<td>2400</td>
</tr>
<tr>
<td>250</td>
<td>10000</td>
<td>8.808</td>
<td>0.499</td>
<td>0.036</td>
<td>21000</td>
</tr>
<tr>
<td>300</td>
<td>10000</td>
<td>14.015</td>
<td>0.459</td>
<td>0.056</td>
<td>28200</td>
</tr>
</tbody>
</table>

1: protein size 2: number of cycles 3: final value of RMSD 4: final value of Restrictions $\times 100$ 5: final value of ScoreLDE 6: total time of the process

Fig. 3: In red the original protein with 100 atoms and in green the protein produced by $SPE_{SW}$ with $C = 6,000$.

Fig. 4: Values of tests results evaluation on a protein structure with 25 atoms, using PM, $C = 6,000$ and $p = 2i + 10$.


