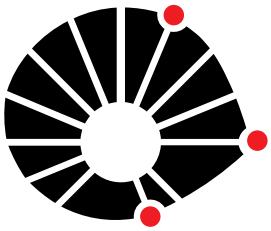

UNICAMP

UNIVERSIDADE ESTADUAL DE CAMPINAS

Instituto de Matemática, Estatística e Computação Científica

ANA FLÁVIA DA CUNHA LIMA

Distance Geometry and the Calculation of Protein Structures using NMR data

Geometria de Distâncias e o Cálculo de Estruturas de Proteínas usando dados de RMN

Campinas

2021

Ana Flávia da Cunha Lima

Distance Geometry and the Calculation of Protein Structures using NMR data

Geometria de Distâncias e o Cálculo de Estruturas de Proteínas usando dados de RMN

Tese apresentada ao Instituto de Matemática, Estatística e Computação Científica da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Doutora em Matemática Aplicada.

Thesis presented to the Institute of Mathematics, Statistics and Scientific Computing of the University of Campinas in partial fulfillment of the requirements for the degree of Doctor in Applied Mathematics.

Supervisor: Carlile Campos Lavor

Co-supervisor: João Eloir Strapasson

Este exemplar corresponde à versão final da Tese defendida pelo aluno Ana Flávia da Cunha Lima e orientada pelo Prof. Dr. Carlile Campos Lavor.

Campinas

2021

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Ana Regina Machado - CRB 8/5467

L628d

Lima, Ana Flávia da Cunha, 1991-
Distance geometry and the calculation of protein structures using NMR data / Ana Flávia da Cunha Lima. – Campinas, SP : [s.n.], 2021.

Orientador: Carlile Campos Lavor.
Coorientador: João Eloir Strapasson.
Tese (doutorado) – Universidade Estadual de Campinas, Instituto de Matemática, Estatística e Computação Científica.

1. Geometria de distâncias. 2. Ressonância magnética nuclear. 3. Análise combinatória. 4. Algoritmos branch-and-prune. I. Lavor, Carlile Campos, 1968-. II. Strapasson, João Eloir, 1979-. III. Universidade Estadual de Campinas. Instituto de Matemática, Estatística e Computação Científica. IV. Título.

Informações para Biblioteca Digital

Título em outro idioma: Geometria de distâncias e o cálculo de proteínas usando dados de RMN
Palavras-chave em inglês:
Distance geometry
Nuclear magnetic resonance
Combinatorial analysis
Branch-and-prune algorithms
Área de concentração: Matemática Aplicada
Titulação: Doutora em Matemática Aplicada
Banca examinadora:
Carlile Campos Lavor [Orientador]
Loana Tito Nogueira
Michael Ferreira de Souza
Cristiano Torezzan
Rafael Santos de Oliveira Alves
Data de defesa: 29-07-2021
Programa de Pós-Graduação: Matemática Aplicada

Identificação e informações acadêmicas do(a) aluno(a)
- ORCID do autor: https://orcid.org/0000-0001-9008-8541
- Currículo Lattes do autor: http://lattes.cnpq.br/7907101925137475

Tese de Doutorado defendida em 29 de julho de 2021 e aprovada

pela banca examinadora composta pelos Profs. Drs.

Prof(a). Dr(a). CARLILE CAMPOS LAVOR

Prof(a). Dr(a). CRISTIANO TOREZZAN

Prof(a). Dr(a). LOANA TITO NOGUEIRA

Prof(a). Dr(a). MICHAEL FERREIRA DE SOUZA

Prof(a). Dr(a). RAFAEL SANTOS DE OLIVEIRA ALVES

A Ata da Defesa, assinada pelos membros da Comissão Examinadora, consta no SIGA/Sistema de Fluxo de Dissertação/Tese e na Secretaria de Pós-Graduação do Instituto de Matemática, Estatística e Computação Científica.

Para minha mãe, minhas irmãs, Matilda e Frajola.

Acknowledgements

Gostaria de agradecer ao meu orientador Carlile e meu co-orientador João pelo auxílio e pelas lições ensinadas e minha família pelo apoio, especialmente a minha mãe que serviu de inspiração para que eu obtivesse meu título de Doutora.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001. It as also partially financed by CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico), process number $168807/2018 - 01$.

Resumo

O objetivo desse trabalho é a determinação de estruturas espaciais de proteínas. Para atingir esse objetivo, o caso Não Associado do Problema de Geometria de Distâncias será apresentado. Em sua forma original, os dados de entrada do Problema de Geometria de Distâncias são um conjunto de vértices (pontos) e uma lista de distâncias associadas a pares desses vértices.

Soluções para instâncias do Problema De Geometria de Distâncias Associado podem ser encontradas através do algoritmo Branch-and-Prune (BP), apresentado na referência (LAVOR et al., 2018). Nesse trabalho, a contribuição principal é uma versão modificada desse algoritmo com o objetivo de atacar o caso Não Associado do problema. Com o objetivo de diminuir cálculos é utilizada uma estratégia de dividir e conquistar baseada no trabalho de Hendrickson na referência (HENDRICKSON, 1995).

Palavras-chave: geometria de distâncias. ressonâcia nuclear magnética. combinatória. branch-and-prune

Abstract

This work aims to determine the spatial structures of proteins. In order to achieve this goal, the Unassigned case of the Distance Geometry Problem (uDGP) will be presented. In its original form, the Distance Geometry Problem has its input data as a set of vertices (points) and a list containing distances associated to vertex pairs.

Solutions for instances of the Assigned Distance Geometry Problem can be found using the Branch-and-Prune (BP) algorithm, presented in reference (LAVOR et al., 2018). In this work, the main contribution is a modified version of this algorithm developed aiming to tackle the Unassigned case of the problem. In an attempt to diminish calculations is used an approach of divide-and-conquer based on the work made by Hendrickson in reference (HENDRICKSON, 1995).

Keywords: distance geometry. nuclear magnetic resonance. combinatorics. branch-and-prune

List of Figures

List of Algorithms

Contents

Introduction

Proteins are a class of nitrogenous organic compounds responsible for many essential activities in living organisms. These molecules can be found in the form of antibodies, enzymes and structural components that support cell structures and fibers that constitute human body muscles, among others.

An example is the prion protein (or PrPc), which acts on the cells of some animals. When its spatial structure is altered it can generate a modified protein called prion scrapie (or PrPSc), as described in reference (LANSBURY; CAUGHEY, 1996) and shown in figure 1. This altered protein has the same composition as its original counterpart, but due to this structural change it causes a disease denominated Bovine Spongiosiform Encephalopathy, also known as the Mad Cow Disease.

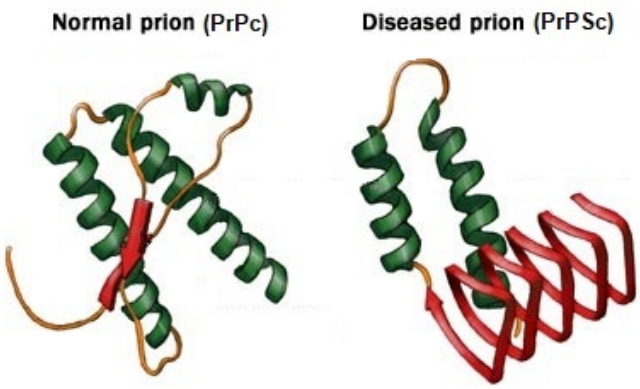

Figure 1 – Isomers of the Prion protein (reference (EDUCATION; (MFMER),)).

This example demonstrates the importance of determining proteins' spatial structures. It is a major problem in biochemistry, since the function of a protein is related not only to its chemical composition but is also strongly linked to its spatial structure (as stated in references (MUCHERINO; LAVOR; LIBERTI, 2010), (CRIPPEN; HAVEL, 1988)). Pursuant to this idea, this study is focused on the problem of determining 3D protein structures using data obtained by Nuclear Magnetic Resonance (NMR) data.

The determination of structures through NMR measures inter-atomic distances of hydrogen atoms present in molecules, providing a list of distances that can be used as input data when trying to find its spatial configuration. It allows proteins to be studied in solution, unlike X-ray crystallography (reference (LAVOR et al., 2018)). This last mentioned technique was the first method developed for the determination of protein structures, and as its name states, it reveals spatial structures of solid crystals.

Proteins are molecules composed by chains of amino acids, forming a pattern

(as seen in figure 2) which will be explored when trying to find its spatial structure. An example of a protein structure can be seen in figure 3.

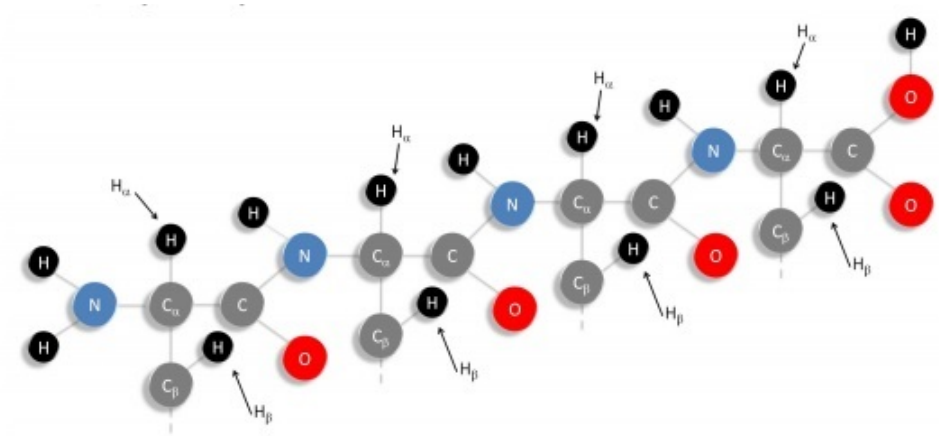

Figure 2 – Protein and its amino acid chain (reference (LAVOR et al., 2012)).

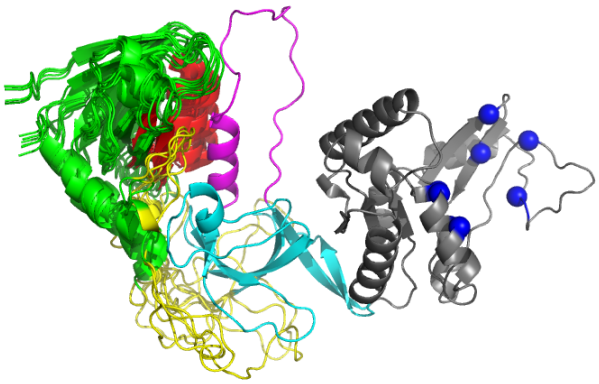

Figure 3 – Example of a protein (reference (UNCCH, 2018)).

In order to achieve this goal the Unassigned case of the Distance Geometry Problem (uDGP) will be presented. In its original form, the Distance Geometry Problem (also referred in this context by the Assigned Distance Geometry Problem or aDGP) has its input data as a set of vertices and a list containing distances associated to vertex pairs. These two elements refer, respectively, to the atoms of the analyzed protein and real, non negative numbers coupled with the atom pairs to which they belong to. Its objective is to find the spatial coordinates of a set of points (also called a *realization*).

Solutions for instances of the Assigned Distance Geometry Problem can be found using the Branch-and-Prune (BP) algorithm, presented in reference (LAVOR et al., 2018). In this work, the main contribution is a modified version of this algorithm is developed aiming to tackle the Unassigned case of the problem. The major modification is done by testing all possible available distances of the input data and the respective candidates they generate for the coordinates of the protein atoms.

This modified version also associates the input distances to pairs of atoms as the atoms themselves are given coordinates. The process of associating coordinates to elements (in this case, atoms) of the problem is also called *realization* (as described in reference (LAVOR et al., 2018)).

The strategy of testing distances generates a larger number of candidates and extends the search space of the Unassigned problem (compared to its Assigned counterpart). In an attempt to diminish calculations is used an approach of divide-and-conquer based on the work made by Hendrickson in reference (HENDRICKSON, 1995). In that work, the idea was to identify rigid substructures in molecules and group its atoms. A rigid body has six degrees of freedom in $\mathbb{R}^3$, but considered independently each vertex has three - therefore, by treating a set of vertices collectively the original problem can be simplified.

Disregarding rotations, translations and reflections, this approach makes it possible to get the coordinates of only one atom in each said substructure and use it as a reference to place all atoms of its group. Using this idea, rigid substructures of the problem's graph are detected in this work in a way that they can all be realized at once instead of one by one, therefore making calculations shorter.

In chapter 1 characteristics of proteins' structures and a model will be presented. In chapter 2 the Distance Geometry Problem will be formally introduced along with the concept of rigidity and their relations to each other. In chapter 3 the definition of a vertex order will be given and it will be shown how it is applied to the problem in this work. In chapter 4 the instance generation process will be presented. In chapter 5 the original Branch-and-Prune algorithm will be presented, and finally, in chapter 6 the modified version of the Branch-and-Prune algorithm will be explained, and results related to its implementation will be shown in chapter 7.

1 Structures and Modeling of Proteins

In this chapter proteins' characteristics will be presented in order to introduce the model that will be used further in the work. The concept of graphs will also be introduced, along with the Backbone Graph, which will be used to model proteins, generate instances for the Unassigned Distance Geometry Problem and appliance of the Branch-and-Prune algorithm.

1.1 Proteins' Structure

Proteins are molecules composed by structures called amino acids. These amino acids are linked to each other, creating a pattern. They are constituted by two types of chains: a main chain that is common to every structure of this type and a side chain that is unique to each particular type of protein. The main chain is composed of atoms of Hydrogen, Carbon, Nitrogen and Oxygen. The side chain has up to 15 atoms linked to each other (as explained in reference (A VALADARES NF, 2006)).

In figure 4 H, N, C and O represent, respectively, hydrogen, nitrogen, carbon and oxygen atoms. G_{SC} represents the side chain distinguishing different amino acids, C_α is the carbon atom connected to G_{SC} and H_α is the hydrogen atom connected to C_α. The links represent bonds between atoms, in line with reference (LAVOR; LIBERTI; MUCHERINO, 2013).

In figure 5 a protein is shown with its chain of three amino acids, where G_{SC_i}, $i = 1, 2, 3$ represent their respective side chains and in figure 6 two amino acids are shown with its side chains highlighted.

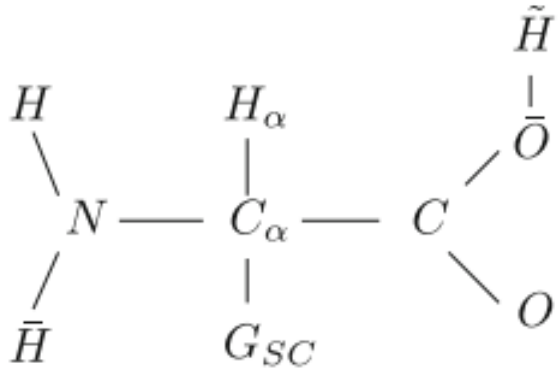

Figure 4 – Amino acid structure (reference (LAVOR; LIBERTI; MUCHERINO, 2013)).

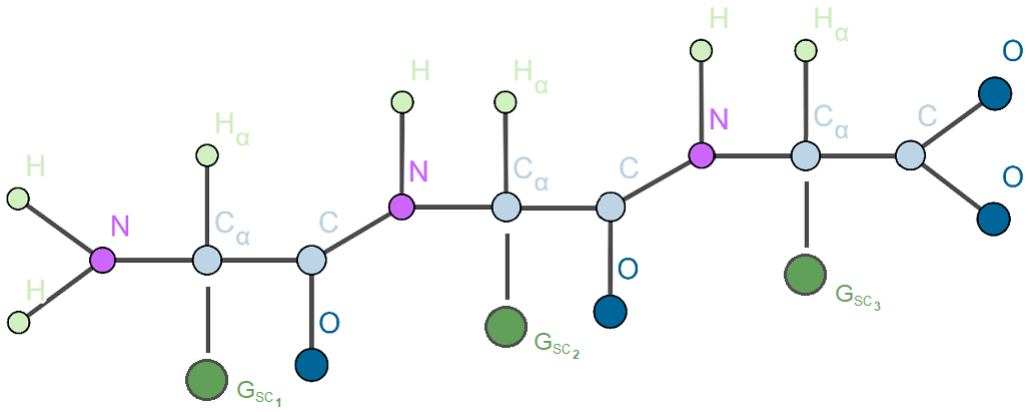

Figure 5 – Protein consisting of three amino acids and their side chains.

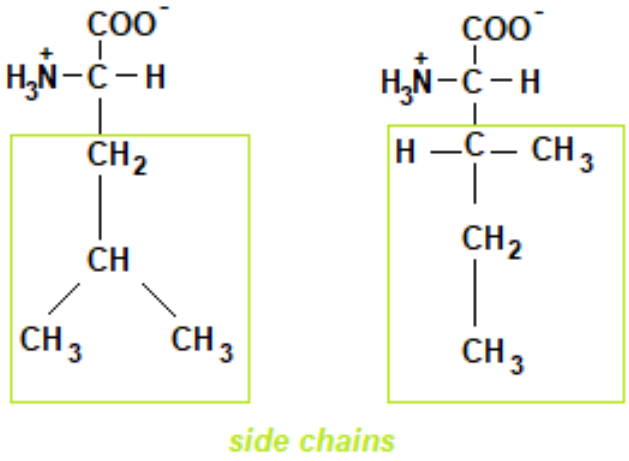

Figure 6 – Representation of two amino acids with their side chains highlighted.

As was also explained in reference (LAVOR et al., 2018), in this work the amino acids' side chains are disregarded and the focus will be the main chain. This way, it is still possible to get a good result while the model remains relatively simple in terms of execution time and computational complexity. This results in the parts regarding G_{SC} in figure 4 and G_{SC_1}, G_{SC_2} and G_{SC_3} in figure 5 being removed from the model. In figure 7 the main chain of a protein is shown.

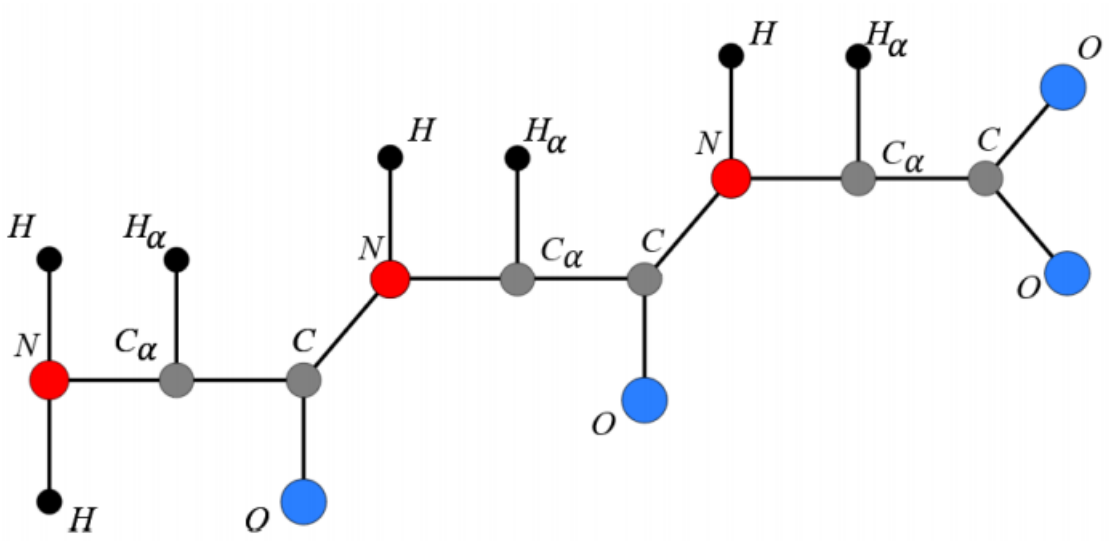

Figure 7 – Representation of a protein's main chain (reference (LAVOR et al., 2018)).

An important characteristic in a protein structure is the *peptide plane*, shown in figure 8. The atoms in this section all lie in the same plane in space due to the peptide bond (highlighted in the figure) that occurs between one of the carbons of one amino acid and the nitrogen of the next amino acid. This is a very stable covalent double bond, which causes the atoms in the peptide plane to be linked in a stronger way than the other atoms in proteins (reference (SCIENCE, 2008)).

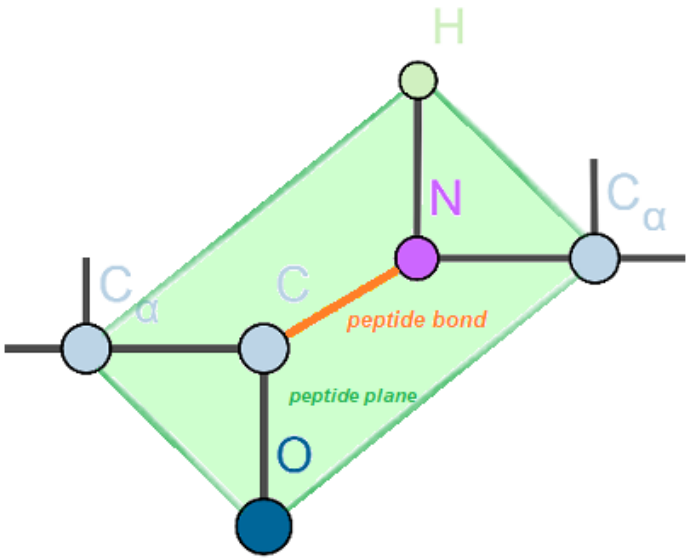

Figure 8 – Detail of a peptide plane in a protein chain.

1.2 Graphs

Definition 1. *(WEST, 2001) A* ***graph*** $G = (V, E)$ *is a composition of two sets:* $V \neq \varnothing$ *(its vertices) and* $E = \{v_i, v_j\} | v_i, v_j \in V, i, j \leqslant |V|$ *(its edges).*

In this work a protein is considered as a set of atoms linked by segments, originally introduced in reference (BROWN, 1865) and also described in reference (LAVOR et al., 2018). While what will be visualized as the end result will be the main chain of the protein and the chemical bonds of its atoms, another type of graph is needed to model these molecules.

1.2.1 The Backbone Graph

The Backbone Graph is a particular kind of graph used to model a protein chain. This model is assembled as chains of atoms constituting a graph $G = (V, E, d)$ composed of:

- A set of vertices V representing the atoms
- A set of edges $E = \{v_i, v_j\} | v_i, v_j \in V$ representing pairs of atoms related to known distances
- A function $d : E \rightarrow [0, \infty)$ associating elements from E to non negative real numbers (distances)

In addition, the following concepts will also be used, illustrated in the figures 9 and 10:

- r_{v_i,v_j}: distance between atoms v_i and v_j having a covalent bond
- θ_{v_i,v_k}: angle between three atoms v_i, v_j, v_k where v_i, v_j and v_j, v_k have a covalent bond
- ω_{v_i,v_l}: angle between the planes formed by the atoms v_i, v_j, v_k and v_j, v_k, v_l where v_i e v_l are separated by three covalent bonds
- x_{v_i}: spatial coordinates of atom v_i

In order to associate values to r_{v_i,v_j}, θ_{v_i,v_k} and ω_{v_i,v_l} the **Geometric Rigidity Hypothesis** is used in the model for this work. This hypothesis, presented in reference (GIBSON; SCHERAGA, 1997), states that it can be assumed that r_{v_i,v_j} and θ_{v_i,v_k} are fixated without having the complexity of the problem altered.

By fixating these values the atoms' coordinates in this model are determined solely by the torsion angles ω_{v_i,v_j}. From here on out, the following variables and values will be used for all v_i, v_j, v_k: $r = 1.526$ Å and $\theta = 1.91$ *rad*. In line with reference (LAVOR; LIBERTI; MUCHERINO, 2013), the Backbone Graph's edges will be defined for all amino acids in the protein chain $i \in \{1, ..., p\}$ (where p is the number of amino acids that compose the protein).

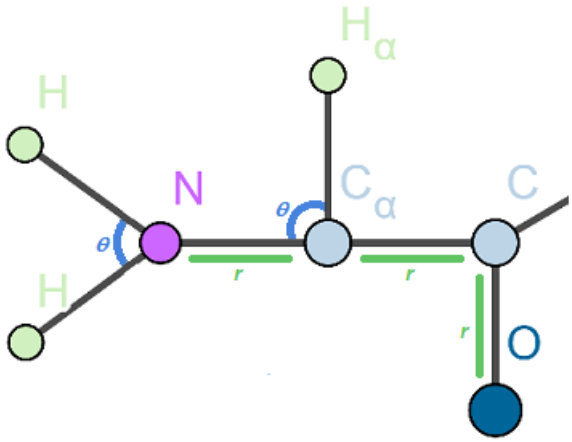

Figure 9 – Distances and angles in a protein's main chain.

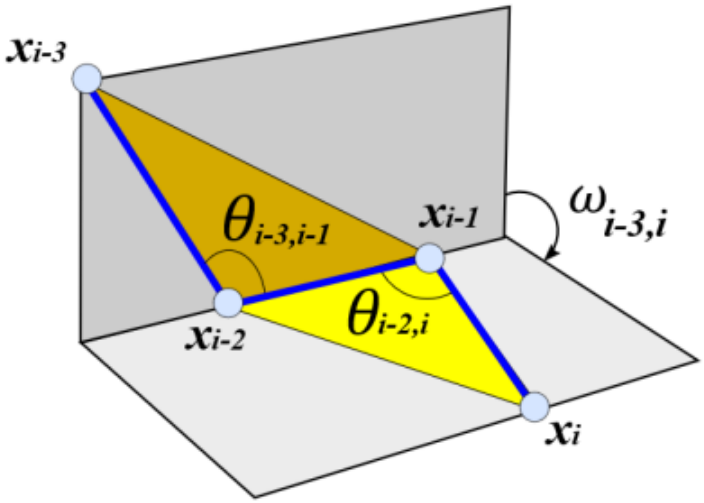

Figure 10 – Illustration of distances and angles in the Backbone Graph (reference (LAVOR et al., 2018)).

Continuing in line with reference (LAVOR; LIBERTI; MUCHERINO, 2013), the process of protein synthesis where a sequence of amino acids bind together will be described by the following graph operations: let

- H, N, C and O be - respectively - hydrogen, nitrogen, carbon and oxygen atoms

- G_{SC} the subgraph representing an amino acid side chain
- C_α be a carbon atom bound to G_{SC}
- H_α be the hydrogen atom bound to C_α
- G'_1 be the graph associated to the first amino acid
- G'_2 be the graph associated to the second amino acid
- $G_{12} = (V_{12}, E_{12})$ be the graph representing two bound amino acids as a result of the following operations (depicted in figure 11):
  - the contraction of $G'_1[\{C, O, \bar{O}, \tilde{H}\}]$ to a vertex labelled C^1 resulting in a modified graph $G_1 = (V_1, E_1)$
  - the contraction of $G'_2[\{\tilde{H}, N\}]$ to a vertex labelled N^2 resulting in a modified graph $G_2 = (V_2, E_2)$
  - $V_{12} = V_1 \bigcup V_2$
  - $E_{12} = E_1 \bigcup E_2 \bigcup \{C^1, N^2\}$

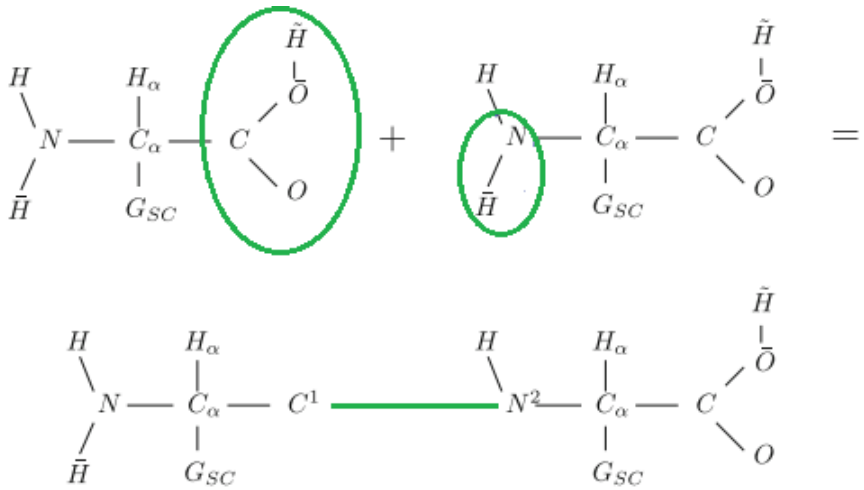

Figure 11 – The binding of two amino acids (reference (LAVOR; LIBERTI; MUCHERINO, 2013)).

As was also stated in reference (LAVOR; LIBERTI; MUCHERINO, 2013), replacing G_1 by G_{12} makes clear the fact that the same operation can be carried out again recursively any finite number $p \in \mathbb{N}$ of times. If this operation is repeated for all the amino acids forming a protein, the resulting graph $G_{12...p}$ with edge set $E_{12...p}$ encoding the covalent bonds represents the whole protein.

It is not uncommon to have more information (*i.e.*, more distances) than what was presented until now, which facilitates the resolution of the original problem. For the

moment, however, only the minimum set of edges needed in the model will be used, and the method to create the Backbone Graph shown ahead can be extended in case more distances are available.

Each symbol $\{H^i, C^i, N^i, O^i, C^i_\alpha, H^i_\alpha\}$ represents the respective atom in the i^{th} amino acid ($i = 1, ..., p$). This will be the vertex set:

$$V_{PB} = \{H^0, H^1, N^1, C^1_\alpha, H^1_\alpha, C^1, ..., H^i, N^i, C^i_\alpha, H^i_\alpha, C^i, ..., H^p, N^p, C^p_\alpha, H^p_\alpha, C^p, O^p, O^{p+1}, H_f\}$$

The pair composed by two symbols ($\{C^i_\alpha, H^i_\alpha\}$ for example) represents an edge in the graph. Using this nomenclature, the following edge sets can be defined:

- $\overline{E}^i_T = \{\{H^i, C^i_\alpha\}, \{N^i, H^i_\alpha\}, \{N^i, C^i\}, \{H^i_\alpha, C^i\}\}\ \forall i \in \{1, ..., p\}$
- $E^1_T = \overline{E}^1_T \cup \{\{H^0, H^1\}, \{H^0, C^1_\alpha\}, \{C^1_\alpha, N^2\}, \{C^1, H^2\}, \{C^1, C^2_\alpha\}\}$
- $E^p_T = \overline{E}^p_T \cup \{\{C^p_\alpha, O^p\}, \{C^p_\alpha, O^{p+1}\}, \{C^p, H_f\}, \{O^p, O^{p+1}\}, \{O^{p+1}, H_f\}\}$
- $E^i_T = \overline{E}^i_T \cup \{\{C^i_\alpha, N^{i+1}\}, \{C^i, H^{i+1}\}, \{C^i, C^{i+1}_\alpha\}\}$

The first item defines the edge set for all amino acids, except for one extra hydrogen in the beginning and one extra oxygen at the end of the chain. The second item includes this hydrogen, the third, the oxygen, and the last item makes the connections between consecutive amino acids. Lastly, the entire edge set can be determined as

$$E_{PB} = E_{12...p} \cup \bigcup_{i \leqslant p} E^i_T$$

and with all these definitions at hand the **Backbone Graph** G_{PB} of a protein, shown in figure 12, can be represented as

$$G_{PB} = (V_{PB}, E_{PB})$$

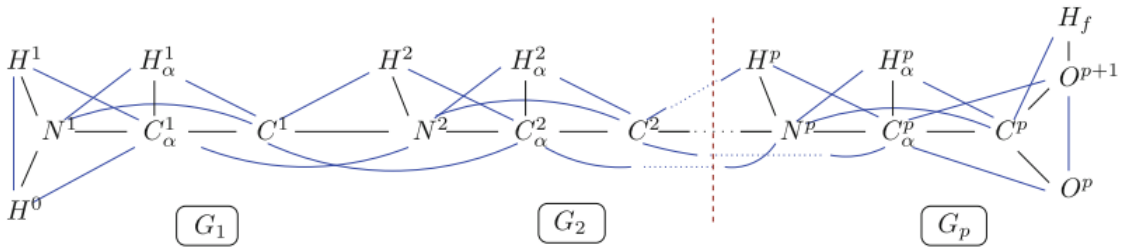

Figure 12 – Backbone Graph associated with a protein composed of three amino acids (minimum set of edges - reference (LAVOR; LIBERTI; MUCHERINO, 2013)).

2 The Distance Geometry Problem

According to reference (LAVOR et al., 2018), Distance Geometry (DG) investigates problems related to the concept of **distance** from a geometric perspective. This area currently focuses on determining a set of coordinates in a geometric space given a set of known distances.

In reference (MENGER, 1928), Menger characterized several geometric concepts based on the notion of distance in 1928, and this is considered the offset of the area. In 1953 Distance Geometry became officially a new research field with the work of Blumenthal (reference (BLUMENTHAL, 1970)).

Initially focused on the determination of whether symmetric matrices are distance matrices, this area had the first explicit expression of its main problem defined in the first paragraph of reference (YEMINI, 1978), and the first work relating Distance Geometry to protein conformation is described in reference (CRIPPEN; HAVEL, 1988).

The following definition for the *Distance Geometry Problem (DGP)* can be found in references (LAVOR et al., 2018) and (LIBERTI et al., 2014a):

Definition 2. ***Distance Geometry Problem (DGP)*** *Given an integer $K > 0$ and a simple, non-directed graph $G = (V, E, d)$ whose edges weights are given by a non-negative function $d : E \rightarrow [0, \infty)$, find a function $x : V \rightarrow \mathbb{R}^K$ such that*

$$\forall \{v_i, v_j\} \in E, ||x_{v_i} - x_{v_j}|| = d_{v_i,v_j} \tag{2.1}$$

where $x_{v_i} = x(v_i)$, $x_{v_j} = x(v_j)$, $d_{v_i,v_j} = d(\{v_i, v_j\})$ and $||x_{v_i} - x_{v_j}||$ is the Euclidean distance between x_{v_i} and x_{v_j}.

Since the focus of this work is protein conformation, here K is set to 3. In reference (SAXE, 1980) it is proved that the problem in an Euclidean space of dimension K is NP-Hard by reducing this problem to the problem known as *3-Satisfability Problem*.

There is still a very important part in the definition of DGP that was not properly explained: the x function. This function maps the vertices of the graph in a Distance Geometry Problem to coordinates in the $\mathbb{R}^K$ ($\mathbb{R}^3$ in this work) space.

Definition 3. *(LAVOR et al., 2018) Given a graph $G = (V, E, d)$, the associated function $x : V \rightarrow \mathbb{R}^K$ is denominated a **realization** of G in $\mathbb{R}^K$*

Definition 4. *If x satisfies all the equations in system 2.1, then x is a **valid realization**.*

Definition 5. *The pair (G, x), where x is a valid realization, is a **framework**.*

The most intuitive approach is to solve the set of equations 2.1 in definition 2. Unfortunately, this process is relatively difficult to do, since this system is not linear and there is evidence that a closed form for its solution may not exist for all cases (reference (BAJAJ, 1988)).

Another common approach is to formulate the Distance Geometry Problem as a global nonlinear optimization problem. In order to do this the decision variables are defined as being the coordinates of the vertices of the structure, so that the coordinate of vertex v_i is given by x_{v_i}. The term to be minimized will then be the difference between the distances of x_{v_i} and x_{v_j}, $\{v_i, v_j\} \in E$ and the "true" distances d_{v_i,v_j} given as part of the input data of the problem. The model is then given by the following formulation:

$$\min_{x_{v_1},\ldots,x_{v_n}\in\mathbb{R}^3} \sum_{\{v_i,v_j\}\in E} (||x_{v_i} - x_{v_j}||^2 - d_{v_i,v_j}^2)^2, \ |V| = n \tag{2.2}$$

However, this approach does not perform well due to the presence of many local minima, as seen in references (LAVOR; LIBERTI; MACULAN, 2006) and (LIBERTI et al., 2014a).

2.1 Rigidity and Distance Geometry

Rigidity plays an important role in the process of finding solutions for DGP instances and asserting the cardinality of its solution set (reference (LAVOR et al., 2018)). Due to its importance, a brief introduction to this subject will be given in this section.

Initially, for the purpose of describing how Rigidity and Distance Geometry are connected, it is necessary to introduce concepts related to frameworks in a Distance Geometry Problem. This is due to the fact that analyzing the (different types of) rigidity of frameworks relates to comparing frameworks having the same graph but different coordinates. Intuitively speaking, such comparison will analyze the distance between the vertices of both frameworks and verify if they are the same, no matter what the coordinates are.

Definition 6. *(GRAVER; AMERICA, 2001) Two frameworks (G, x) and (G, y) are **isometric**, denoted by $(G, x) \sim (G, y)$, if*

$$\forall \{v_i, v_j\} \in E, ||x_{v_i} - x_{v_j}|| = ||y_{v_i} - y_{v_j}|| \tag{2.3}$$

Definition 7. *(GRAVER; AMERICA, 2001) Two frameworks (G, x) and (G, y) are **congruent**, denoted by $(G, x) \equiv (G, y)$, if*

$$\forall v_i, v_j \in V, ||x_{v_i} - x_{v_j}|| = ||y_{v_i} - y_{v_j}|| \tag{2.4}$$

Definition 8. *(GRAVER; AMERICA, 2001)* (G, x) *is a* ***rigid framework*** *if there is a real number* $\epsilon > 0$ *such that*

$$(G, y) \sim (G, x) \text{ and } ||x_{v_i} - y_{v_i}|| < \epsilon, \forall v_i \in V \Rightarrow (G, y) \equiv (G, x) \tag{2.5}$$

Equation 2.5 in definition 8 says that a rigid framework has no continuous deformations other than rotations and translations, as stated in reference (GRAVER; AMERICA, 2001). In this sense, each deformation would count as a different framework having the same originating graph.

Solutions for the DGP constitute a set that will be denoted by X and can be rotated and translated in $\mathbb{R}^3$, implying that this set is infinite and uncountable. Using algebraic geometry it is also possible to prove that the cardinality of X (assuming it is not an empty set) is either finite or uncountable (as was done in reference (BENEDETTI; RISLER, 1990)), a result strongly related to rigidity (explained in reference (GRAVER; AMERICA, 2001)).

For the next definition the following will be considered: $G = (V, E)$, a graph with $|V| = n$ and $|E| = m$, and (G, x) a framework in $\mathbb{R}^3$. Let also $R\lambda = 0$ be a linear system where $\lambda \in \mathbb{R}^{3n}$ and R is a $m \times 3n$ matrix where each row with index $\{v_i, v_j\}$ has exactly 6 nonzero entrances given by

$$x_k(v_i) - x_k(v_j) \tag{2.6}$$

and

$$x_k(v_i) - x_k(v_j) \tag{2.7}$$

for all $\{v_i, v_j\} \in E$ and $k = 1, 2, 3$, with $x_1(v_i)$, $x_2(v_i)$ and $x_3(v_i)$ being the (x, y, z) Cartesian coordinates of x_{v_i} in $\mathbb{R}^3$.

Definition 9. *(LIBERTI et al., 2014a) The framework* (G, x) *is infinitesimally rigid if the only solutions for* $R\lambda = 0$ *are translations and rotations.*

Theorem 1. *(GRAVER; AMERICA, 2001), (CONNELLY, 1987) Infinitesimal rigidity implies rigidity.*

If it is assumed that the input data is accurate (that is, the given distances are real numbers and not intervals), the DGP solution set will contain all solutions to the problem that are compatible with the input distances. This means that the more distances are available, the smaller the number of solutions is (excluding rotations, translations and reflections) since more constraints to the solutions are added. If all distances between vertex pairs are given there is only one solution, which can be found in linear time (reference (DONALD, 2011)).

In this work and in many others the concept of a *rigid graph* will be used in the sense of a graph G having a rigid framework (G, x). A characterization of all rigid graphs in $\mathbb{R}^2$ is given by Laman in reference (LAMAN, 1970) but unfortunately the problem of characterizing all rigid graphs in dimension 3 and above is still open. However, there are a few conjectures available, some of which (specially regarding graphs with frameworks in $\mathbb{R}^3$) can be found in reference (JACKSON; JORDÁN, 2008).

2.2 The Assigned and Unassigned Classes

The Distance Geometry Problem (DGP) is divided into two subcategories: Assigned (aDGP) and Unassigned (uDGP) (as stated in references (LIBERTI et al., 2014a), (DUXBURY L. GRANLUND, 2016)). In the Assigned case the distances are given assigned to vertex pairs, and in the Unassigned case only the distances are given, without the vertex pairs to which they belong to.

Therefore, in the Unassigned case it is also necessary to associate the input distances to vertices pairs (in addition to finding the realization for the vertices). In the case of distances given by NMR it is not known which pairs of vertices they belong to, which implies that the problem in this work falls into the Unassigned case. The formal definitions for these two classes are given below.

Definition 10. ***Assigned Distance Geometry Problem (aDGP)*** *(DUXBURY et al., 2021) Given an integer $K > 0$ and an undirected simple graph $G = (V, E, d)$ whose edges have weights given by a non-negative function $d : E \to [0, \infty)$, determine whether there exists a function $x : V \to \mathbb{R}^K$ such that*

$$\forall \{v_i, v_j\} \in E, ||x_{v_i} - x_{v_j}|| = d_{v_i, v_j} \tag{2.8}$$

where $x_{v_i} = x(v_i)$, $x_{v_j} = x(v_j)$, $d_{v_i,v_j} = d(\{v_i, v_j\})$ and $||x_{v_i} - x_{v_j}||$ is the Euclidean distance between x_{v_i} and x_{v_j}.

Suppose now that only a list of distances $\hat{d}$ of size m is given and let L be the index set of $\hat{d}$, ie, $L = \{1, 2, ..., m\} \subset \mathbb{N}$. Then $\hat{d} = (d_l, d_2, ..., d_m) \in \mathbb{R}^m_+$.

Definition 11. ***The Unassigned Distance Geometry Problem (DGP)*** *(DUXBURY et al., 2021) : given a set of vertices V and a list of distance values $\hat{d} = d_1, ..., d_m$, find an injective function $g : \{1, ..., m\} \to V \times V$ and a function $x : V \to \mathbb{R}^3$ such that, $\forall \{i, j\} \in g(\{1, ..., m\})$,*

$$||x_{v_i} - x_{v_j}||_2 = d_{v_i v_j} \tag{2.9}$$

and

$$d_{v_i v_j} = \hat{d}_{g^{-1}(\{i,j\})} \tag{2.10}$$

As it was previously mentioned, in the case approached in this work the input does not have a graph since it is not possible to get its edges. This is due to the fact that NMR only provides distances, not the atoms related to these distances. Therefore, at the same time that the vertices are realized they are also associated with the distances received in the input data of the problem, composing the problem graph.

3 Vertex Orders

As stated in reference (BODLAENDER et al., 2012; MUELLER; MARTIN; LUMSDAINE, 2007; HENNEBERG, 1886), problems related to graphs and its vertex orders are intimatelly related subjects. The problem of determining a graph rigidity based on an order for its vertices was originally introduced in reference (HENDRICKSON, 1992).

Given a vertex $v \in V$ in a graph $G = (V, E)$ its adjacent vertices have a strong influence on the cardinality of the solution set of the DGP: as it was previously said, if there is a small number of adjacent vertices the number of solutions can become uncountable.

An example of the importance of vertex orders are graph instances having all vertices starting from the fifth being adjacent to at least four already realized vertices (called a *trilateration order*). A graph whose vertices can form an order having this property is not only globally rigid but also has an unique incongruent solution which can be found in linear time (references (LAVOR et al., 2018), (GIBSON; SCHERAGA, 1997), (LIBERTI et al., 2014a)).

According to references (LAVOR et al., 2012), (LAVOR et al., 2018), although instances for the DGP do not usually have a trilateration order, information related to the chemistry of proteins can be used to find a different order. This new order has close ties to the way the problem is resolved, specially with the Branch and Prune algorithm.

In the next chapters it will be seen that in each iteration of this algorithm the coordinates of a vertex are found using three vertices prior to it who have already been realized. In order for this to be feasible, a vertex order that guarantees that every instance for the problem can have its vertices ordered in such way is necessary.

The vertex order ultimately used in the BP algorithm is called the *hand-crafted order*, introduced in reference (LAVOR et al., 2018). The purpose of this chapter is to present this order, starting with precursory concepts necessary for its grasp: the more general formal definition of vertex orders, the *discretizable distance geometry* problem class deriving from instances having such order, and a specific type of vertex order denominated *re-orders*.

Definition 12. *(LAVOR et al., 2012), (LAVOR et al., 2018) A* ***vertex order*** *for a graph $G = (V, E, d)$ is a sequence $r : N \to V$ of length $|r| \in N$ such that*

- *The first three vertices of G constitute a clique:*

$$\{v_1, v_2\}, \{v_1, v_3\}\ \{v_2, v_3\}\ \in E \tag{3.1}$$

- *All vertices starting from the fourth are adjacent to at least three predecessors:*

$$\forall i > 3, \exists j, k, l \mid j < i, k < i, l < i : \{v_j, v_i\}, \{v_k, v_i\}, \{v_l, v_i\} \in E \tag{3.2}$$

Instances with the properties just described in definition 12 define a subcategory of the Distance Geometry Problem:

Definition 13. *(LAVOR et al., 2018), (LAVOR et al., 2012), (GONÇALVES; MUCHERINO, 2014) The instance class whose elements have vertices that follow the order described in definition 12 is denominated the* ***Discretizable Distance Geometry Problem (DDGP)***

In reference (LAVOR et al., 2012) an algorithm to find all incongruent solutions of instances was presented called Branch and Prune. This algorithm is the starting point of the Branch-and-Prune algorithm developed in this work, which will be presented ahead.

3.1 Re-Orders

The natural idea for an order in the vertex set of a graph is to trace a simple path with no cycles. Unfortunately, this strategy won't satisfy the second part of definition 12 for DDGP instances. It is possible, however, to satisfy this order definition if vertex repetitions are allowed, and in reference (LAVOR; LIBERTI; MUCHERINO, 2013) the concept of repetition orders is introduced:

Definition 14. *Given a graph $G = (V, E, d)$ a* ***repetition order*** *(re-order) is a sequence $r : \mathbb{N} \to V$ of length $|r| \in \mathbb{N}$ such that*

- *The first three vertices of G constitute a clique:*

$$\{v_1, v_2\}, \{v_1, v_3\} \{v_2, v_3\} \in E$$

- *All vertices as from the fourth are adjacent to at least three predecessors:*

$$\forall i > 3, \exists j, k, l \mid j < i, k < i, l < i : \{v_j, v_i\}, \{v_k, v_i\}, \{v_l, v_i\} \in E$$

- *For all $i \in \{4, ..., |r|\}$ either $v_{i-3} = v_i$ or $\{v_{i-3}, v_i\} \in E$.*

In practical terms the first property means that the distances $d_{v_1,v_2}, d_{v_1,v_3}, d_{v_2,v_3} \in (0, \infty)$ - that is, these distances are real numbers greater than zero. The second property translates to $d_{v_i,v_{i-1}}, d_{v_i,v_{i-2}} \in (0, \infty)$ for $i = 4, ..., |r|$, ie, these distances must also be real numbers greater than zero (and cannot be intervals, as it is otherwise permitted in some points in these definitions).

For the third property there are three possibilities:

- $d_{v_i,v_{i-3}} = 0$ - there is a vertex repetition
- $d_{v_i,v_{i-3}} \in (0, \infty)$ - that is, v_{i-3} and v_i are separated by one, two or three chemical bonds and the given distance for the pair is exact
- $d_{v_i,v_{i-3}} = [\underline{d}_{v_i,v_{i-3}}, \overline{d}_{v_i,v_{i-3}}] | 0 < \underline{d}_{v_i,v_{i-3}} < \overline{d}_{v_i,v_{i-3}}$ - the distance is not exact, but an interval

In this work there will only be used instances that have exact distances. An algorithm to find solutions for instances of the DMDGP having inexact distances can be found in reference (LAVOR; LIBERTI; MUCHERINO, 2013) and approaches for this instance class of the problem utilizing Clifford Algebra were also develped in references (ALVES; LAVOR, 2017), (R. et al., 2017).

Repetitions are a way to access the input data in a way that all vertices of the problem can be realized. The initial clique guarantees that all solutions found are incongruent and the strict triangular inequality assures that the cardinality of the solution set is countable, according to reference (LAVOR et al., 2018).

Defining repetition orders also sets a minimum necessary quantity and form of input data for the problem that agrees with the way the data is extracted since NMR usually not only provides the minimum set of distances but also extra distances that can be used to accelerate the search for a solution. The coordinates of a repeated vertex can also be recalculated in order to prevent numerical instabilities.

Whereas repetition is a great tool for finding solution for instances of the DMDGP, one has to be careful when using it. Even with the assurance of the strict triangular inequality, it is still possible to find infinite possibilities for the following vertex to be realized. One example of such case is three consecutive vertices in the sequence where two of them are equal.

The geometrical correspondence of a case like this is a triangle having one of its sides equal to zero, drifting away from the idea of restricting its possible positions to points in space (reference (LAVOR et al., 2018)). In order to prevent this issue, repetitions can only occur in vertices $\{v_i, v_j\}$ such that $|i - j| \geqslant 3$, as stated in the third restriction in definition 14.

In the construction process for the backbone graph it can be seen that it has repetitive parts that represent the amino acids composing a protein: all backbone graphs have characteristic initial and final structures and a middle structure that repeats itself according to its size.

With this in mind, it was formulated an order that can be used in any graph of this type called *re-orders*, developed in reference (LAVOR; LIBERTI; MUCHERINO, 2013). Here the already familiar denomination is being used once more, where a protein

atom is a vertex of the backbone graph and its exponent indicates its correspondent amino acid: C^2, for example, indicates the carbon atom of the second amino acid.

Firstly, an order is associated to the first half of the first amino acid in the G_{PB} graph (denoted by r^1_{PB}). This assures the first requirement of a re-order is satisfied, since N^1, H^1 and H^0 constitute a clique. Moreover, all following vertices have their three predecessors already realized. This is illustrated in figure 13:

$$r^1_{PB} = \{N^1, H^1, H^0, C_\alpha, N^1, H^1_\alpha, C^1_\alpha, C^1\}$$

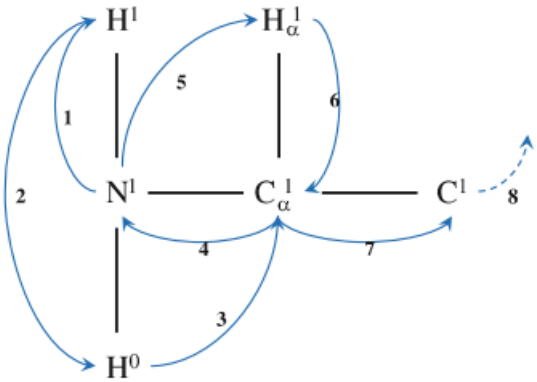

Figure 13 – Order r^1_{PB} (reference (LAVOR; LIBERTI; MUCHERINO, 2013)).

The following part of the order (denominated r^2_{PB}) corresponds to atoms of the first amino acid's second half. Again, the repetition of atoms N^1 and C^1_α (and atoms N^i and C^i_α in the next part) assures the fulfillment of the second and third requirements in definition 14. It is shown in figure 14:

$$r^2_{PB} = \{N^1, H^1, H^0, C^1_\alpha, N^1, H^1_\alpha, C^1_\alpha, C^1\}$$

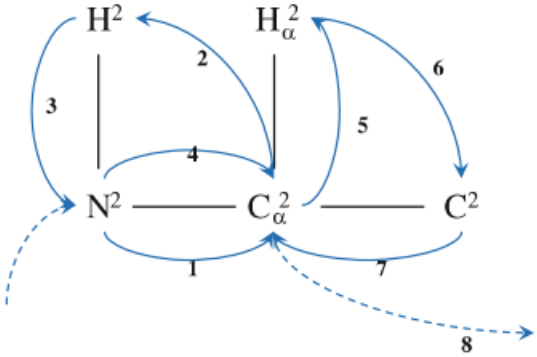

Figure 14 – Order r^2_{PB} (reference (LAVOR; LIBERTI; MUCHERINO, 2013)).

As described in reference (LAVOR; LIBERTI; MUCHERINO, 2013), this atom sequence is used to construct a bridge between the first and second parts of the first amino acid. From this point on the following order (shown in figure 15) repeats itself, in line with the characteristic pattern of the backbone graph:

$$r^i_{PB} = \{N^i, C^{i-1}, C^i_\alpha, H^i, N^i, C^i_\alpha, H^i_\alpha, C^i, C^i_\alpha\}$$

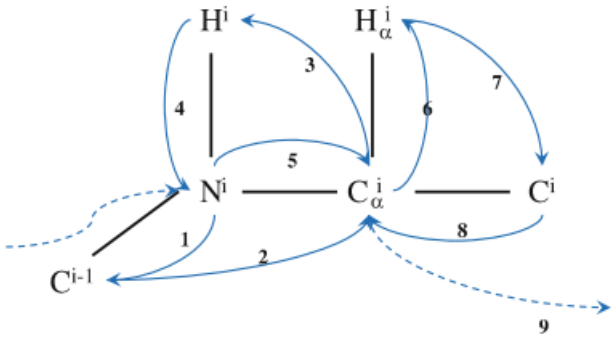

Figure 15 – Generic order r^i_{PB} (reference (LAVOR; LIBERTI; MUCHERINO, 2013)).

The chain's last amino acid also has a different structure in its last half, having a few extra atoms. In this case the corresponding order is the following (shown in figure 16):

$$r^p_{PB} = \{N^p, C^{p-1}, C^p_\alpha, H^p, N^p, C^p_\alpha, H^p_\alpha, C^p, C^p_\alpha, O^p, C^p, O^{p+1}\}$$

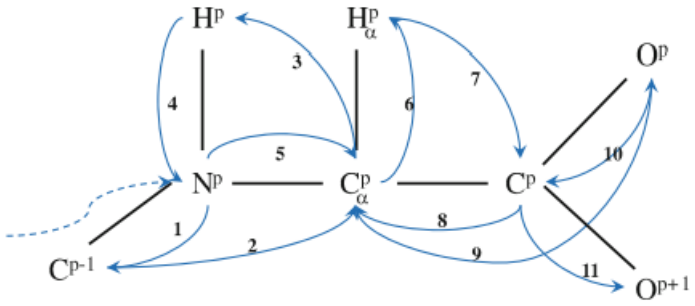

Figure 16 – Last part of the re-order r^p_{PB} (reference (LAVOR; LIBERTI; MUCHERINO, 2013)).

3.2 The Hand-Crafted Vertex Order

The hand-crafted vertex order is introduced in reference (LAVOR et al., 2018), where it is presented following a formal definition of a protein graph and the backbone of a protein.

For the latter, let p be the number of amino acids in a protein and, for $k = 2, ..., p-1$:

- N^k the nitrogen atom in the k^{th} amino acid
- O^k the oxygen atom in the k^{th} amino acid
- C_α^k the α-carbon atom bonded to the H_α^k atom in the k^{th} amino acid
- C^k the carbon atom bonded to O^k atom in the k^{th} amino acid
- H_α^k the α-hydrogen atom in the k^{th} amino acid
- H^k the hydrogen atom bonded to N^k in the k^{th} amino acid

And since the extremities of the chain have each one atom more, with an extra hydrogen atom at the beginning and an extra oxygen atom at the end, they are labeled H^{1*} and O^{p*} respectively. In the hand-crafted order both of these atoms will always appear after their counterpart, with H^{1*} right after H^1 and O^{p*} being placed last. An example of a protein backbone is shown in figure 17 for $p = 3$. In figure 18 the same structure is presented with its respective atoms labeled according to this denomination.

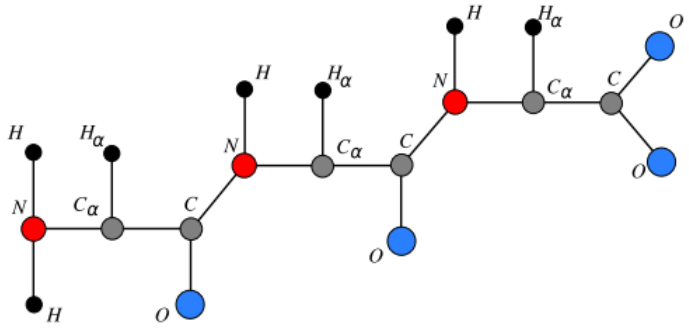

Figure 17 – Protein backbone (reference (LAVOR et al., 2018)).

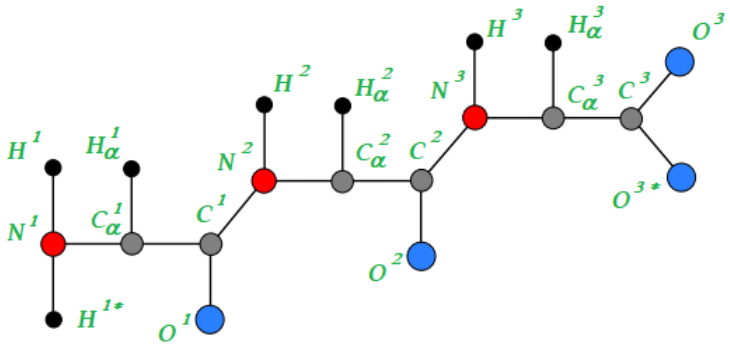

Figure 18 – Protein backbone labeled according to the protein's backbone denomination (reference (LAVOR et al., 2018)).

With these denominations, a protein's backbone can be defined as

$$\{N^k, C_\alpha^k, C^k\} \tag{3.3}$$

and the graph associated to this protein backbone will be denoted as

$$G = (V, E, d) \tag{3.4}$$

where, as stated in reference (LAVOR et al., 2018), $k = 2, ..., p-1$.

The final result is the following order:

$$\begin{gathered} hc = \{N^1, H^1, H^{1'}, C_\alpha^1, N^1, H_\alpha^1, C^1, C_\alpha^1, ..., \\ H^i, C_\alpha^i, O^{i-1}, N^i, H^i, C_\alpha^i, N^i, H_\alpha^i, C^i, C_\alpha^i, ..., \\ H^p, C_\alpha^p, O^{p-1}, N^p, H^p, C_\alpha^p, N^p, H_\alpha^p, C^p, C_\alpha^p, O^p, C^p, O^{p'}\} \end{gathered} \tag{3.5}$$

and an example for an instance having three amino acids is available in figure 19.

In reference (LAVOR et al., 2018) it is proven that the hand-crafted order is a re-order. This is achieved, firstly, observing the fact that since it is assumed that all bond lengths and bond angles are fixed at their equilibrium values (in line with definition 1.2.1 - the rigid geometry hypothesis), the first and the second requirements of a re-order (definition 14) are satisfied.

Lastly, it has to be shown that the hand-crafted order fulfills the third requirement in the definition of re-orders. This last requirement is equivalent to stating that either the distances between vertices v_i and v_{i-3} are always known or vertices v_i and v_{i-3} are the same.

Since this order repeats itself, it is easy to isolate all possible atom pairs for vertices v_i and v_{i-3}. Using chemical and structural properties such as the rigid geometry hypothesis and the peptide plane, it is shown that these distances can always be calculated and therefore the demonstration that the hand-crafted order is a re-order is concluded.

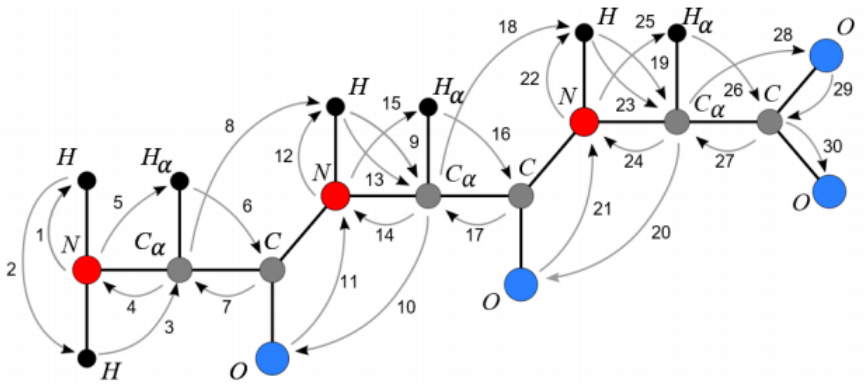

Figure 19 – Hand-crafted order (reference (LAVOR et al., 2018)).

4 Instance Generation

The process of instance generation is a useful tool for testing the Branch-and-Prune algorithm. It also helps understanding in-depth how proteins are modeled, since the closer to the reality the model used is, better are the results obtained.

An instance in the context of this work is regarded as the input data for both the original version and the modified version of the Branch-and-Prune algorithm. In the original version, after the protein's atoms are given labels (usually like $1, 2, 3, ...$ etc) a distance value is associated to atom pairs. In the modified algorithm version of this work, only a distance list of real nonnegative values is provided.

4.1 Lavor Instances

In order to facilitate further descriptions (and as was done in reference (PHILLIPS; ROSEN; WALKE, 1994)), the following sets regarding atoms in a protein will be defined:

- M_1 is the set of all pairs of consecutive atoms v_i, v_j
- M_2 is the set of all pairs of atoms separated by two covalent bonds v_i, v_k
- M_3 is the set of all pairs of atoms separated by three covalent bonds v_i, v_l
- M_4 is the set of all pairs of atoms separated by more than two covalent bonds v_i, v_j

The Lavor Instance's model, introduced in reference (LAVOR, 2006), considers each atom pair $v_i, v_j \in M_1$ as having a bond length $r_{v_i v_j}$ corresponding to the Euclidean distance between x_{v_i} and x_{v_j}. Every $v_i, v_j, v_k \in M_2$ has a bond angle $\theta_{v_i v_k}$. This bond angle is measured according to the relative position of atom v_k with respect to the line γ containing atoms v_i and v_j. Similarly, every $v_i, v_j, v_k, v_l \in M_3$ has a torsion angle $\omega_{v_i v_l}$ measured according to the angles between the planes constituted by atoms v_i, v_j, v_k and v_j, v_k, v_l.

In line with the Rigid Geometry Hypothesis (definition 1.2.1), in reference (LAVOR, 2006) all bond lengths and bond angles are fixed as $r_{v_i v_j} = 1.526$ Å $\forall v_i, v_j \in M_2$ and $\theta_{v_i v_k} = 109.5^\circ$ (or 1.91 rad) $\forall v_i, v_k \in M_3$. Let S be the edge set for the protein graph, defined according to a cut-off value (often being set as 5Å due to the fact that NMR data values are usually less than or equal to it).

Given this cut-off value, the instances are generated and only the values less than or equal to it are taken. This is made so this simulated data becomes more similar to

a possible NMR data. The members of S will then be the pairs of atoms associated to these distances in the instance.

The first step is to obtain the Cartesian coordinates for the atoms. In order to do that, the following equations are used:

$$\begin{bmatrix} x_{v_{i_1}} \\ x_{v_{i_2}} \\ x_{v_{i_3}} \\ 1 \end{bmatrix} = B_1, B_2, ..., B_i \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \qquad \forall i = 1, ..., n$$

$$B_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$B_2 = \begin{bmatrix} -1 & 0 & 0 & -d_{v_1,v_2} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$B_3 = \begin{bmatrix} -\cos\theta_{v_1,v_3} & -\sin\theta_{v_1,v_3} & 0 & -d_{v_2,v_3}\cos\theta_{v_1,v_3} \\ \sin\theta_{v_1,v_3} & -\cos\theta_{v_1,v_3} & 0 & d_{v_2,v_3}\cos\theta_{v_1,v_3} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$B_i = \begin{bmatrix} -\cos\theta_{v_{i-2},v_i} & -\sin\theta_{v_{i-2},v_i} & 0 & -d_{v_{i-1},v_i}\cos\theta_{v_{i-2},v_i} \\ \sin\theta_{v_{i-2},v_i}\cos\omega_{v_{i-3},v_i} & -\cos\theta_{v_{i-2},v_i}\cos\omega_{v_{i-3},v_i} & -\sin\omega_{v_{i-3},v_i} & d_{v_{i-1},v_i}\sin\theta_{v_{i-2},v_i}\cos\omega_{v_{i-3},v_i} \\ \sin\theta_{v_{i-2},v_i}\sin\omega_{v_{i-3},v_i} & -\cos\theta_{v_{i-2},v_i}\sin\omega_{v_{i-3},v_i} & \cos\omega_{v_{i-3},v_i} & d_{v_{i-1},v_i}\sin\theta_{v_{i-2},v_i}\sin\omega_{v_{i-3},v_i} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\forall i = 4, ..., n$$

where n is the number of atoms in the molecule. Once these coordinates are calculated, the distances between each pair of atoms are measured, the ones less or equal to the cut-off value are added to a list. This distance list will be the instance and the input data for the Branch-and-Prune algorithm.

In the case of the Assigned Distance Geometry Problem this list also contains the atom pairs related to these distances. An example with 8 atoms is shown in figure 20, and its set S (given by the v_i and v_j columns) along with the associated distances is in equation 4.1. The cut-off value used was 4Å.

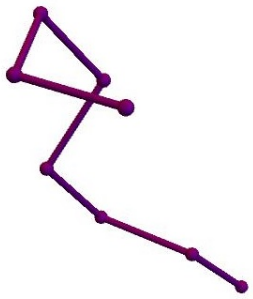

Figure 20 – Example of a protein backbone generated in reference (LAVOR, 2006).

$$
\begin{pmatrix}
v_i & v_j & d_{v_i v_j} \\
1 & 2 & 1.526 \\
1 & 3 & 2.49139 \\
1 & 4 & 2.91598 \\
1 & 5 & 3.56236 \\
2 & 3 & 1.526 \\
2 & 4 & 2.49139 \\
2 & 5 & 2.94969 \\
3 & 4 & 1.526 \\
3 & 5 & 2.49139 \\
3 & 6 & 3.83788 \\
4 & 5 & 1.526 \\
4 & 6 & 2.49139 \\
4 & 7 & 2.94994 \\
5 & 6 & 1.526 \\
5 & 7 & 2.49139 \\
5 & 8 & 3.8359 \\
6 & 7 & 1.526 \\
6 & 8 & 2.49139 \\
7 & 8 & 1.526
\end{pmatrix} \tag{4.1}
$$

5 The Branch-And-Prune Algorithm

The goal of the Branch-And-Prune (BP) algorithm is to find a valid realization $x : V \to \mathbb{R}^3$ (from definition 3) for the atoms of a protein modeled as a backbone graph $G = (V, E, d)$, where the elements of d frequently have a maximum value defined by a cutoff value c. This happens due to the fact that NMR experiments usually perceive distances smaller than a given value (most of the times this value being equal to 5 Å).

The vertices are realized according to the hand-crafted vertex order defined in chapter 3 so that it can be guaranteed that, at every step, there will be enough information to not only find coordinates for these vertices but also find a finite number of possibilities for them.

5.1 Initialization

Assuming the vertices are ordered according to the hand-crafted order, the first step is to realize the first three elements of the sequence (figure 19). This guarantees that these three first vertices form a clique and therefore all distances between pairs of these elements are available.

Since this is all the available information, it is necessary to define deliberately the coordinate of the first vertex v_1, which in this case is the origin of the Euclidean space. A similar argument places the second vertex v_2 at the axis x and the third vertex v_3 at the plane xy. Applying this to system 6.2 the result is:

$$\begin{aligned} x_1 &= (0, 0, 0) \\ x_2 &= (-d_{v_1,v_2}, 0, 0) \\ x_3 &= (-d_{v_1,v_2} + d_{v_2,v_3} \cos\theta_{v_1,v_3}, d_{v_2,v_3} \sin\theta_{v_1,v_3}, 0) \end{aligned} \tag{5.1}$$

where $d_{v_i,v_j} \in d, \{v_i, v_j\} \in E | v_i, v_j \in V$.

5.2 General Procedures

The process continues by realizing the following vertices in the hand-crafted order (presented in section 3.2). This assures the distances between every vertex $v_i \in V$, $i \geqslant 4$ and its three immediate predecessors v_{i-1}, v_{i-2} and v_{i-3} are available. Hence, it is possible to use equations from system 6.2, substituting the respective distances. This makes possible for the algorithm to find coordinates for the remaining vertices.

According to the hand-crafted order, it can also be assured that not only there will always be a solution to system 6.2 but also that the number of solutions for the coordinates of any given vertex in a branch of the search tree will be at most two and therefore every vertex has two possible positions. As a consequence, a binary tree is formed associating possible coordinates for the set of atoms in the protein chain (shown in figure 21).

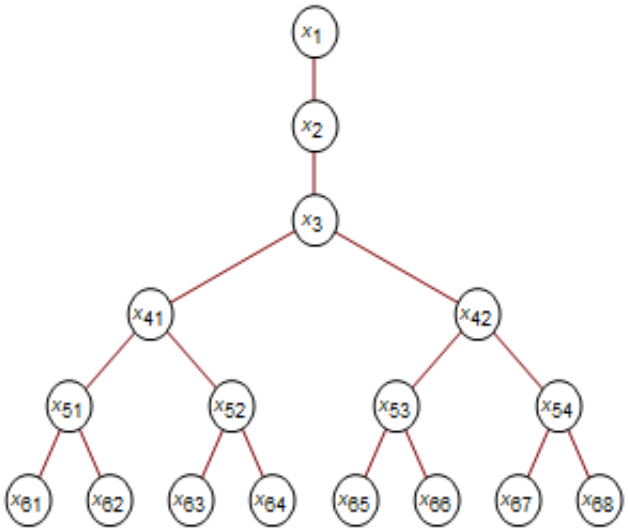

Figure 21 – Binary tree representing the possible solutions for an instance containing 6 vertices.

5.2.1 Prune

As was previously said in section 1.2.1 there is also frequently more information given in the input data than only the distances related to the three immediate predecessors of a vertex. In this case it is possible to use this extra distance to compare to the distance between the points realized regarding these two vertices and verify if they are equal.

In case the distances do not agree, the current structure is not a solution for the instance and therefore there is no need to continue to explore that branch of the binary tree and realize the remaining vertices regarding this specific candidate. This procedure is called *pruning*.

For the cases where the is a cutoff value c for the distances, there may not be comparisons available since the distance between the current realized vertex ans past vertices can be greater than c. When this is the case, nothing is done and the algorithm continues.

The pseudocode for the Branch-and-Prune algorithm is given in algorithm 1.

Algorithm 1 Branch-and-Prune algorithm.

Require: Number of amino acids p and distance function d

```
Find the coordinates for the first three vertices (section 5.1)

Priority line = [ ]
while A solution is not found do
    Generate candidates
    Prune
    if There is at least one candidate then
        Add corresponding rigid substructure to existing chain
        Put last generated candidates at the front of priority line
    Explore next candidate in priority line
```

5.3 Symmetries

In reference (LIBERTI et al., 2014b) it was shown that the number of solutions for DGP instances is not only even but a power of two. This is related to the fact that these solutions are partially symmetric. The characteristic rigidity of the backbone graph implies that the search space is finite and has a cardinality multiple of 2^{n-3}, with n being the number of vertices in the graph (as stated in reference (LAVOR et al., 2018)).

At each point in the BP tree the two points obtained as solutions of system 2 are symmetric. The symmetry plane I (figure 22) is defined by the three predecessors used to realize these points. That way, all points realized from there on will also be symmetric regarding I, and given the fact that distances are preserved in symmetric structures, they will also be pruned in a symmetric manner when viewed as points in the BP tree.

In reference (FIDALGO et al., 2018) a strategy to find all incongruent solutions for an aDGP instance using this property is presented. This way, the Branch-and-Prune algorithm only has to find one solution and keep track of prunes in the partial tree (depicted in figure 23). The remaining solutions are found by partial reflections, giving the result shown in figure 24).

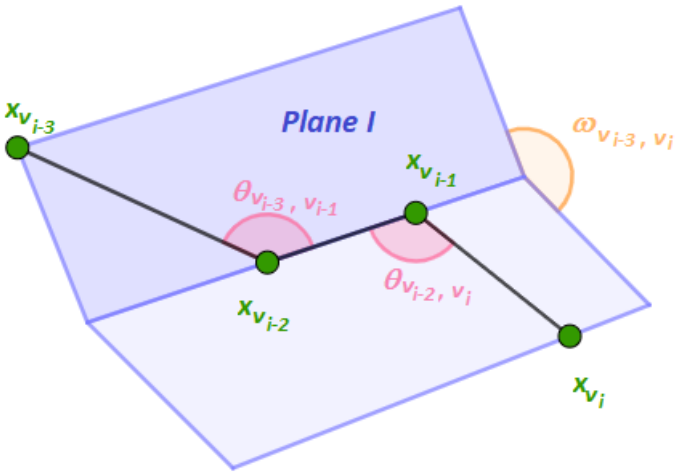

Figure 22 – Plane I.

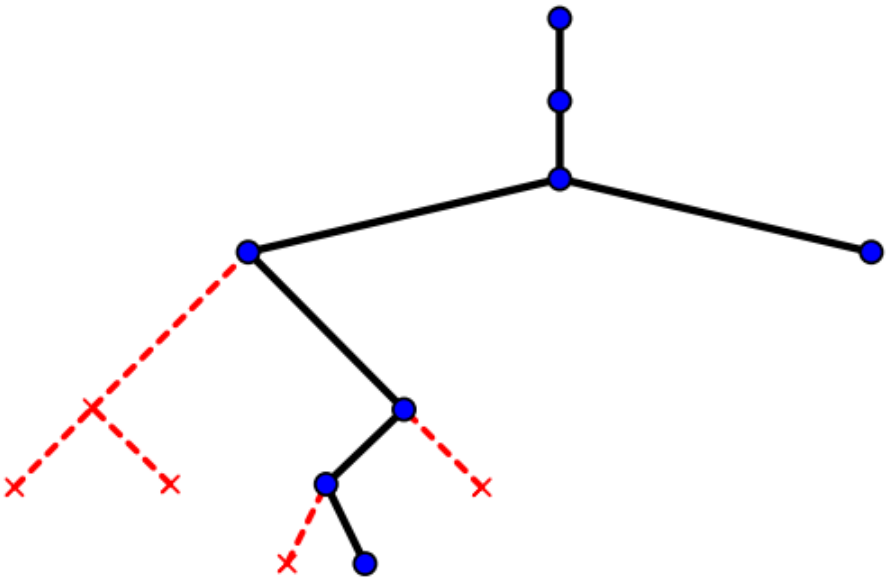

Figure 23 – BP partial tree after the original solution is found (reference (LAVOR, 2014)).

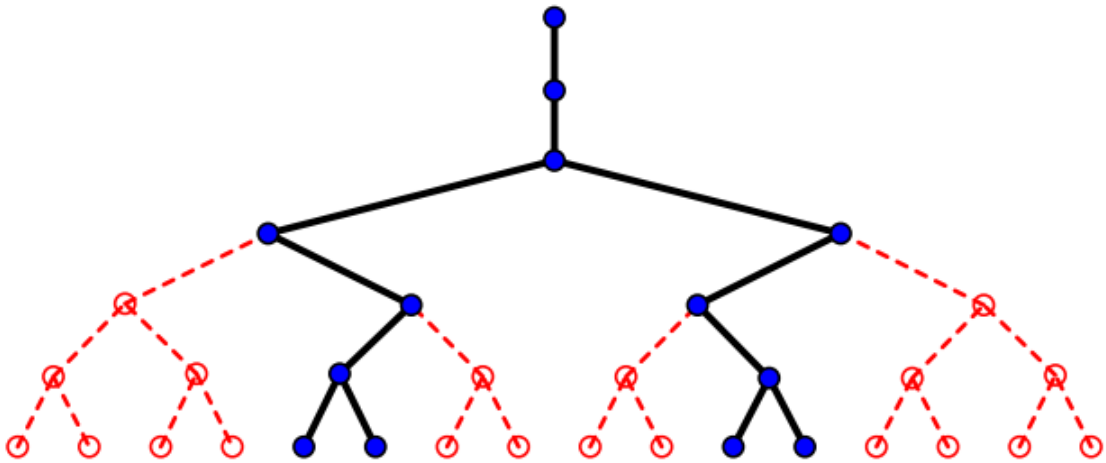

Figure 24 – BP tree symmetries (reference (LAVOR, 2014)).

6 The Modified Branch-And-Prune Algorithm

The original Branch-and-Prune algorithm was developed to find solutions for instances of the Assigned Distance Geometry Problem. Since this work treats the Unassigned version of the problem, it is necessary to take the differences between these versions into consideration. This approach resulted in a modified version of the original Branch-and-Prune algorithm that will be discussed ahead.

While both versions of the Distance Geometry Problem have the same final objective (ie, finding the coordinates for a set of points given distances between pairs of these points), assigning these distances to actual pairs in the set of points is a detail that changes almost every aspect of the problem's approach. The fact that the Unassigned case requires distances and not which edges these distances belong to implies that there is not a graph in the problem input, but a set of vertices (the atoms, in the case of this work's application) and a set of edge weights, but not the edges themselves.

Consequently, addressing this problem includes defining the associated graph, that is, defining the function that associates pairs of vertices (ie the edges). Moreover, it is necessary to associate these edges to the input $\hat{d}$. In practical terms, this means that when the vertices are realized the distances are associated to pairs of vertices, thus defining the edges of the graph.

In figure 25 two instances corresponding to the same solution are depicted. As it can be seen, the Unassigned problem instance contains only distances, while the Assigned problem instance contains distances associated to hydrogen atom pairs (as a reference, figure 26 depicts the solution for these instances along with the reference for these pairs). Since the Assigned problem provides these atom pairs, it also defines the problem graph.

One thing to take into consideration when analyzing these examples is that, while only hydrogen distances are given, the distances between other atom pairs not composed by two hydrogens in the same rigid part (as will be explained ahead in section 6.1.1) are known. This is due to the fact that, as the name states, these sections are rigid and hence their atoms and structure are always the same. It is also the case that, other than the first part of the protein, all rigid substructures have only one hydrogen, therefore making the determination of all distances in each rigid subpart possible.

$$\begin{pmatrix} \{2.49\} & \{2.93\} & \{5.03\} \\ \{7.33\} & \{4.29\} & \{7.41\} \\ \{3.83\} & \{5.27\} & \{6.69\} \\ \{4.08\} & \{6.46\} & \{2.17\} \\ \{4.83\} & \{2.59\} & \{6.23\} \\ \{2.92\} & \{2.67\} & \{5.75\} \\ \{3.70\} & \{4.75\} & \{3.63\} \end{pmatrix}$$

Unassigned Distance Geometry Problem Instance

$$\begin{pmatrix} \{2.49 \to (1,2)\} & \{2.93 \to (1,3)\} & \{5.03 \to (1,4)\} \\ \{7.33 \to (1,5)\} & \{4.29 \to (1,6)\} & \{7.41 \to (1,7)\} \\ \{3.83 \to (2,3)\} & \{5.27 \to (2,4)\} & \{6.69 \to (2,5)\} \\ \{4.08 \to (2,6)\} & \{6.46 \to (2,7)\} & \{2.17 \to (3,4)\} \\ \{4.83 \to (3,5)\} & \{2.59 \to (3,6)\} & \{6.23 \to (3,7)\} \\ \{2.92 \to (4,5)\} & \{2.67 \to (4,6)\} & \{5.75 \to (4,7)\} \\ \{3.70 \to (5,6)\} & \{4.75 \to (5,7)\} & \{3.63 \to (6,7)\} \end{pmatrix}$$

Assigned Distance Geometry Problem Instance

Figure 25 – Example of two instances (Assigned and Unassigned problems) having the same solution.

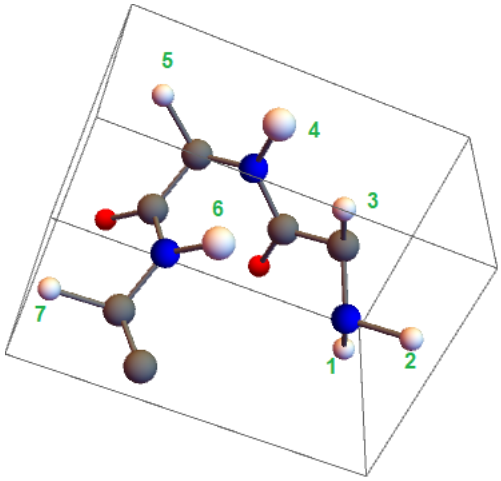

Figure 26 – Corresponding protein structure and hydrogen references for the two instances depicted in figure 25.

In this version of the BP algorithm, a map of the used distances and to which pair of vertices these distances belong to is kept (along with the realization to which that map belongs), for each node of the BP tree. At each iteration all unused distances between the minimum and maximum possible distance values are taken and a list of possible points for the next hydrogen is generated.

6.1 Instance Generation in This Work

In reference (HENDRICKSON, 1995) Hendrickson developed an algorithm based on global optimization that uses the strategy of divide and conquer in order to find the solution for what he defined as the *molecule problem*. This problem seeks to find the localization of a set of objects in the Euclidean space given a set of distances associated to pairs of these objects, which is a definition very similar to the Assigned Distance Geometry Problem.

The Hendrickson algorithm analyzes the input graph and divides it into rigid substructures. This way, only one atom of each rigid substructure has to be realized (in order to be used as an anchor) and the other atoms can be placed according to its relative position regarding the other elements of the set, what can save calculating time.

6.1.1 Rigid Substructures

In line with the original idea of the BP algorithm a modified version was created where one of the objectives was to focus only on the calculation of its hydrogen coordinates. This was possible due to the fact that, according to the premises of the DDGP (from definition 3), the hand-crafted vertex order (section 3.2), properties of the peptide plane (figure 8) and the chirality of the molecule (explained ahead in 6.1.2), all other atoms of a protein have only one possible position.

As was stated in section 2.1, rigid frameworks don't have any continuous deformations other than rotations and translations. Another way to state this is saying that, in a rigid framework, all vertex distances are maintained constant throughout any movement or force applied to it. This implies that, given a rigid framework and all of its vertex pairs' distances, if the position for one of its vertices is fixed then all of the remaining vertex positions can be found (excluding rotations).

In section 1.1 the general structure of proteins is presented, where the peptide plane is explained and shown to have strong atom bonds. These strong bonds allied to the atom placements and the fact that all of these atoms lay in one plane imply that this is a rigid structure. The other structures, being tetrahedrons, naturally have a rigid nature in the Euclidean 3D space.

The model for a protein in this work disregards its side chains, leaving only its main chain. This main chain has a pattern where three substructures (peptide plane and two types of tetrahedrons) are repeated. These substructures will be explained in detail in the next section, and are also rigid. This, along with the assumptions that the instances for the uDGP will have an order like the one described in definition 12 and that the values of the bond angles θ and bond distances r are known and set to their equilibrium values (as the Geometric Rigidity Hypothesis in definition 1.2.1 states) will be the main base for this work development hereon now.

Each rigid substructure described ahead has exactly one hydrogen, except for the first tetrahedron that has two hydrogens. Since the first tetrahedron appears only once at the beginning of the resolution, the calculations will not be affected by this and the algorithm can focus in realizing only the hydrogens and sing its coordinates as an anchor to place multiple atoms at a time in the structure.

The reasons for focusing on the placing of rigid substructures to form the

final backbone are related to an attempt to reduce significant rounding errors as well as decreasing the number of iterations for the algorithm. The idea is to save time, since it is no longer necessary to calculate the coordinates of each atom separately.

Although this approach may seem different from the hand-crafted order, this tool is still being used. For each of the rigid substructures containing exactly one hydrogen, the first element to be accessed using that order is precisely that hydrogen. The difference here is that instead of placing just the hydrogen in its position the entire rigid substructure is placed. This way, all atoms that are not hydrogens can be skipped.

6.1.2 Rigid Substructures Applied to Lavor Instance Generation

Starting from the model proposed by the Lavor Instances (presented in section 4.1) and taking into consideration the Hendrickson approach introduced in reference (HENDRICKSON, 1995) (as was stated in section 6.1), it can be seen that there are three kinds of rigid substructures:

- A tetrahedron formed by two hydrogen atoms, one nitrogen atom and one α-carbon atom (figure 27 and circled in green in figure 30)
- A tetrahedron formed by one hydrogen atom, one nitrogen atom and two carbon atoms (figure 28 and circled in pink in figure 30)
- A peptide plan formed by one hydrogen atom, one nitrogen atom, three carbon atoms and one oxygen atom (figure 29 and circled in blue in figure 30)

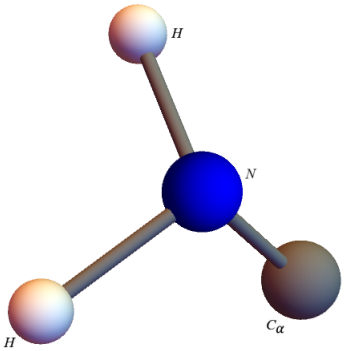

Figure 27 – First tetrahedron.

These substructures have parts in common, and it is precisely what is used when linking them, as seen in figure 31. In figure 30 these rigid substructures are highlighted: the blue circle highlights the peptide plane, the green circle highlights the first tetrahedron and the pink circle highlights the second tetrahedron. It should also be noticed that the amino acids that constitute the molecule are formed by the union of (second) tetrahedron HNCC and the peptide plane, repeating themselves throughout the protein structure.

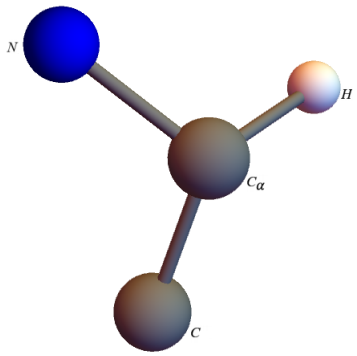

Figure 28 – Second tetrahedron.

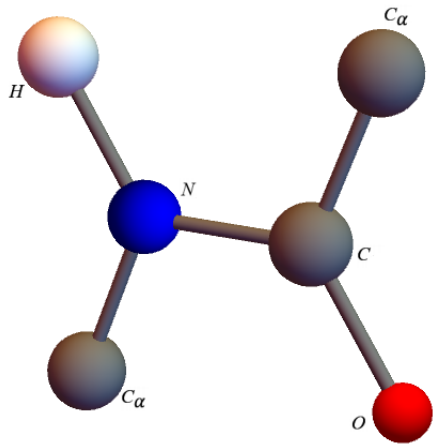

Figure 29 – Peptide plane.

It is also valid to introduce, at this point, the concept of chirality. According to references (DONALD, 2011), (CRIPPEN; HAVEL, 1988), it is a geometrical property of some types of molecules where, given the same composition and atomic connectivity, there are differences in their spacial arrangement of atoms. These different structures are mirrored to each other but are not superimposable. As a comparison, an object such as a sphere doesn't have this property since it is equal to its mirror image. An example of chiral structures if shown in figure 32.

Since a protein's function is heavily associated to its geometry, the ones who can be found in different chiralities often cannot be accounted as being the same protein. For the purpose of this work, the term "positive chirality" will be used for the "right" molecule and "negative chirality" will be used otherwise.

Using the bond length $r = 1.526$Å and bond angle $\theta = 1.91$ rad values (both given by the *Rigid Geometry Hypothesis* in definition 1.2.1), the notions of the rigid structures that compose the model (peptide plane, tetrahedrons) and knowing which

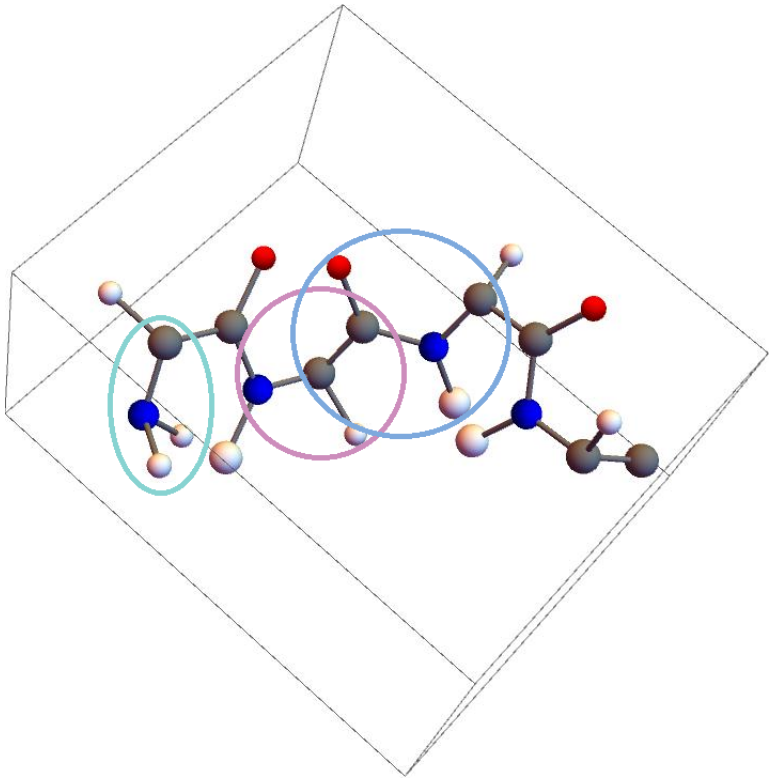

Figure 30 – Highlighted rigid substructures in an instance. Hydrogen atoms are in white, carbon in gray, nitrogen in blue and oxygen in red.

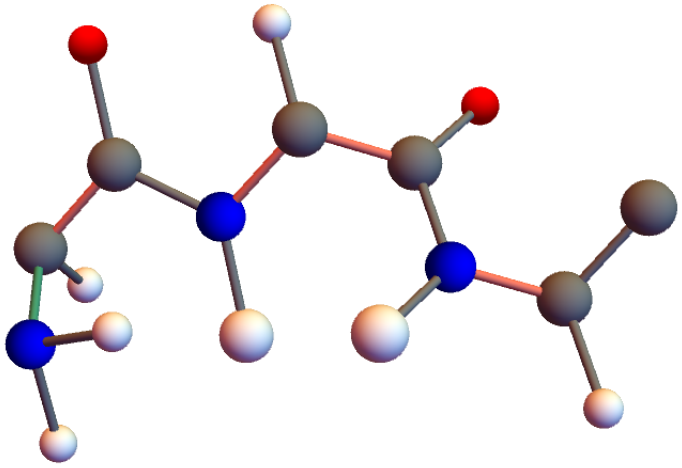

Figure 31 – Common parts in rigid substructures. In green there is the link between the two tetrahedrons and in pink there are the links between the second tetrahedron and the peptide plane.

mirrored molecule it is desired to be found (chirality), it is possible to generate an instance.

The atoms' coordinates of each rigid substructure can be found by ordering said atoms according to the hand crafted order and placing the first atom at the origin, the second in the yz plane and the third in the z line in space (this ordering is shown in figures 35, 36 and 38). After that, the i^{th} atom can be found finding the interception of three

spheres centered in atoms v_{i-1}, v_{i-2} and v_{i-3} and with radii $d_{v_i,v_{i-1}}$, $d_{v_i,v_{i-2}}$ and $d_{v_i,v_{i-3}}$ respectively. After that, it is merely a question of fitting them into the right positions to construct the protein backbone.

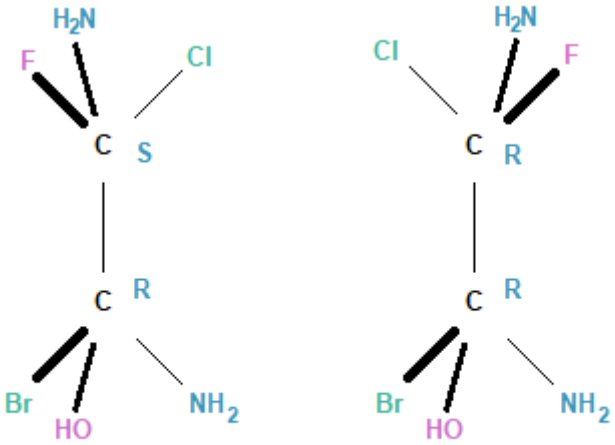

Figure 32 – Chiral structures (reference (FLORENCIO, 2016)).

The configuration of each tetrahedron can be found by placing the first atom in the origin, the second in the x axis, and the third in the xy plane. The coordinates of the fourth atom are given by the intersection of three spheres, each centered on one of the preceding atoms and with radii given by the distance between these atoms and the fourth atom, as shown in figures 33 and 34. The intersection of these spheres gives us two solutions and we choose the correct one using the property of chirality.

Still analyzing the structure of the two types of tetrahedra, there is one atom in the center (nitrogen for the first tetrahedron and α-carbon for the second tetrahedron) and the other three atoms are all the same distance from that center. Thus, these three atoms form an equilateral triangle of side r_2 and angles $\theta_2 = 2\pi/3$. Using the values $r = 1.526$Å and $\theta = 1.91$ radians, the sides of this equilateral triangle (which are the distances between atoms around the center of the tetrahedron) can be calculated using the cosine law:

$$r_2 = (2r^2(1 - \cos\theta))^{1/2} = 2.49139 \tag{6.1}$$

Finally, the coordinates of the atoms can be calculated by placing the first atom at the origin, the second atom in the coordinate $(r, 0, 0)$, the third atom in the coordinate $(r\cos\theta, r\sin\theta, 0)$ and the coordinate p of the fourth atom is found by solving the system

$$\begin{aligned} |(0,0,0) - p| &= r \\ |(r,0,0) - p| &= r_2 \\ |(r\cos\theta, r\sin\theta, 0) - p| &= r_2 \end{aligned} \tag{6.2}$$

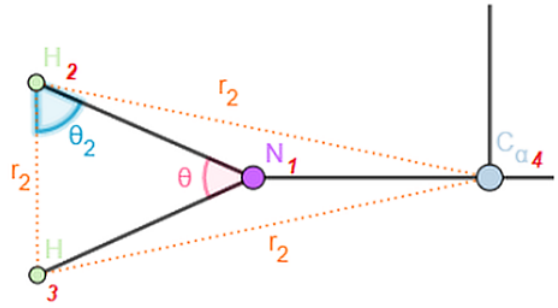

Figure 33 – Distances and angles in the first tetrahedron.

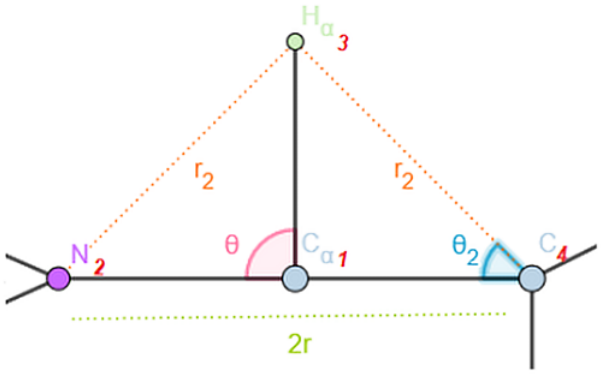

Figure 34 – Distances and angles in the second tetrahedron.

The system 6.2 results in two solutions for the point p and the final solution is chosen according to the chirality of the molecule, such solution being $p = (0.522434, 0.732271, -1.23269)$. This is the model for the origin-centered tetrahedra, and from now on it is only needed to rotate and translate them in alternation with the peptide plane to form the chain of the molecule.

The coordinates of the first tetrahedron are given by:

$$\begin{aligned} x_N &= (0,0,0) \\ x_C &= (-r,0,0) \\ x_{H_1} &= (-r\cos\theta, -r\sin\theta, 0) \\ x_{H_2} &= p \end{aligned} \tag{6.3}$$

and the coordinates of the second tetrahedron are given by:

$$\begin{aligned} x_{C_1} &= (0,0,0) \\ x_N &= (-r,0,0) \\ x_H &= (-r\cos\theta, -r\sin\theta, 0) \\ x_{C_2} &= p \end{aligned} \tag{6.4}$$

and the respective atoms to which those denominations refer to are schematized in the figures 35 and 36.

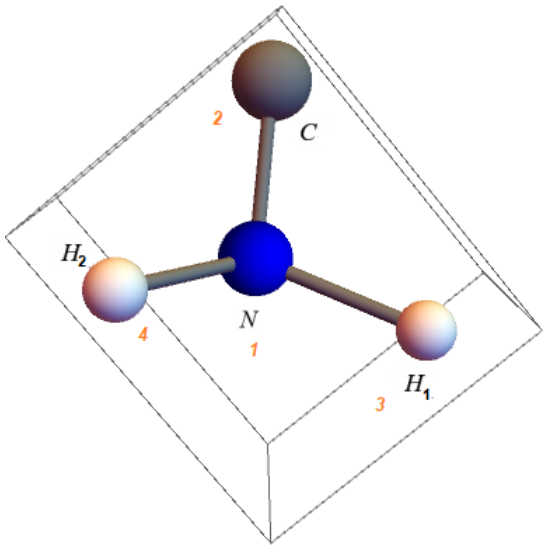

Figure 35 – Reference for the first tetrahedron.

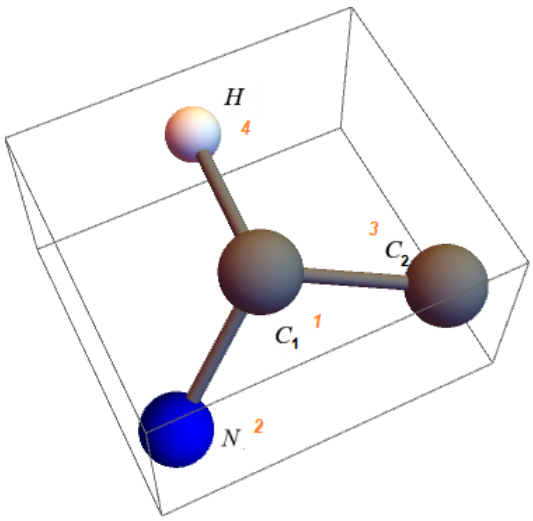

Figure 36 – Reference for the second tetrahedron.

The first atom of the peptide plane (the central carbon denominated by C_1) is again placed at the origin. The rest of the initial procedure of placing the first three atoms is similar to the procedure done in the tetrahedra. The second atom (the nitrogen denominated by N) is placed on the line $yz = 0$, resulting in coordinates equal to $(-r, 0, 0)$ (given the fact that it has a covalent bond with C_1 and therefore its distance to C_1 is equal to r).

The third atom (the carbon that the peptide plane shares with the next tetrahedron denominated C_2) is placed in the $z = 0$ plane. This restrain added to the fact that its distances to N and C_1 and the bond angle between C_1, N e C_2 are equal to equal to r, r_2 and θ respectively results in coordinates equal to $(-r(1 - \cos\theta), -r\sin\theta, 0)$. A reference illustration for these atoms is shown in the figure 37.

In order to allocate the remainder of the atoms it is used the fact that the variables r and θ have known values along with the fact that atoms in a peptide plane are all in same plane. It can then be deduced that, since the third carbon C_3 is bonded to C_1 which in turn is bonded to N, the angle $C_3 - C_1 - N$ is equal to θ. Thus the triangle formed by these three atoms is isosceles (since the distance between C_3 and C_1 equals the distance between C_1 and N given by r). Therefore, the angle $N - C_3 - C_1$ can be deduced to be $\beta = (\pi - \theta)/2$. The details are shown in the figure.

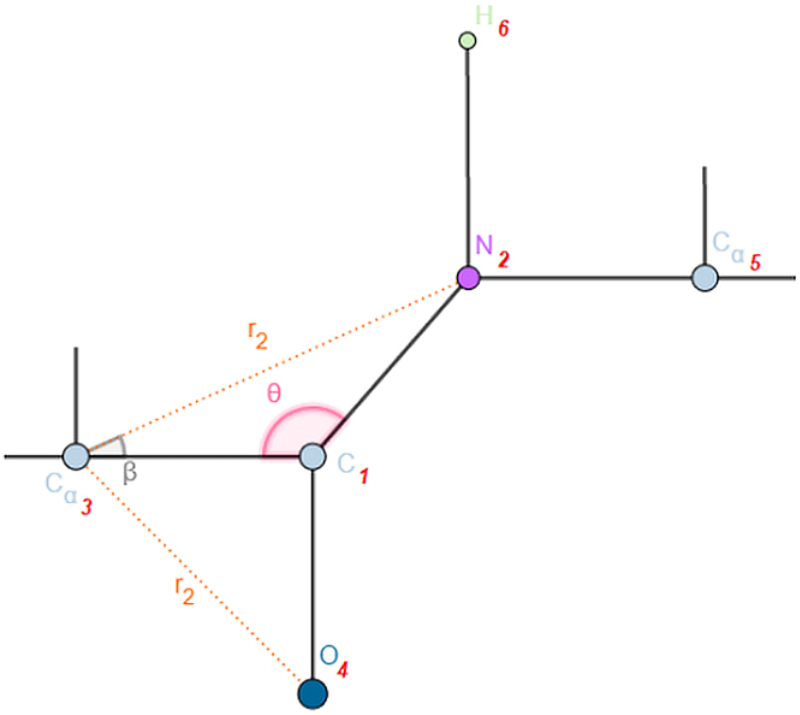

Figure 37 – Distances, angles, and references to the peptide plane.

The list of all the coordinates of the peptide plane centered in the origin is given by:

$$\begin{aligned} C_1 &= (0, 0, 0) \\ N &= (-r, 0, 0) \\ C_2 &= (-r(1 - \cos\theta), -r\sin\theta, 0) \\ O &= (-r\cos\beta, -r\sin\beta, 0) \\ C_3 &= (-r\cos\theta, r\sin\theta, 0) \\ H &= (-r(1 - \cos\beta), r\sin\beta, 0) \end{aligned} \tag{6.5}$$

The description for instance generation is shown in algorithm 2.

In the process there is a cutoff value c inserted in order to mimic the NMR data collection process, which calculates the distances between nearby hydrogens and provides those having value up to c. This value is typically equal to 5 Å in experiments.

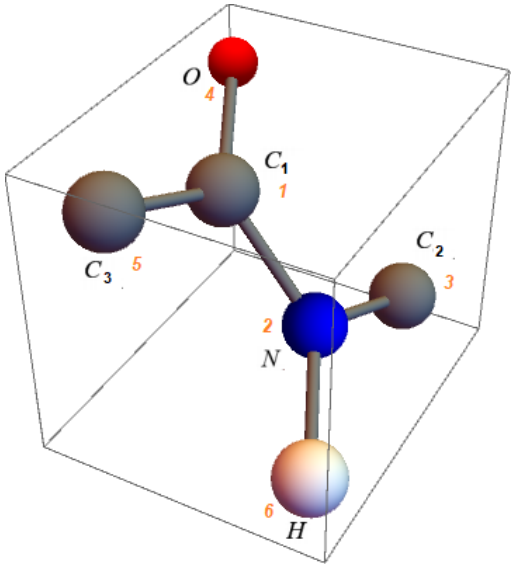

Figure 38 – Reference to the peptide plane.

Algorithm 2 Instance generation algorithm

Require: Number of amino acids p, cutoff value c and distances $\hat{d}$

1: Set up rigid substructures
2: Place tetrahedron 1
3: **for** $i = 2$ to $2p$ **do**
4: Randomly generate torsion angle ω
5: **if** i is even **then**
6: Add tetrahedron 2 to existing chain
7: **else** Add peptide plane to existing chain
8: Rotate the last placed rigid substructure ω rad around the common axe with previous structure

6.2 Initialization (Modified Algorithm)

In order to initiate the algorithm it is firstly necessary to set up the chain's rigid parts. The coordinates for the elements of these rigid substructures with relative positions at the origin are again calculated:

- First tetrahedron $NCHH$ (figure 35)

$$\begin{aligned} x_N &= (0,0,0) \\ x_C &= (-r,0,0) \\ x_{H_1} &= (-r\cos\theta, -r\sin\theta, 0) \\ x_{H_2} &= p \end{aligned} \tag{6.6}$$

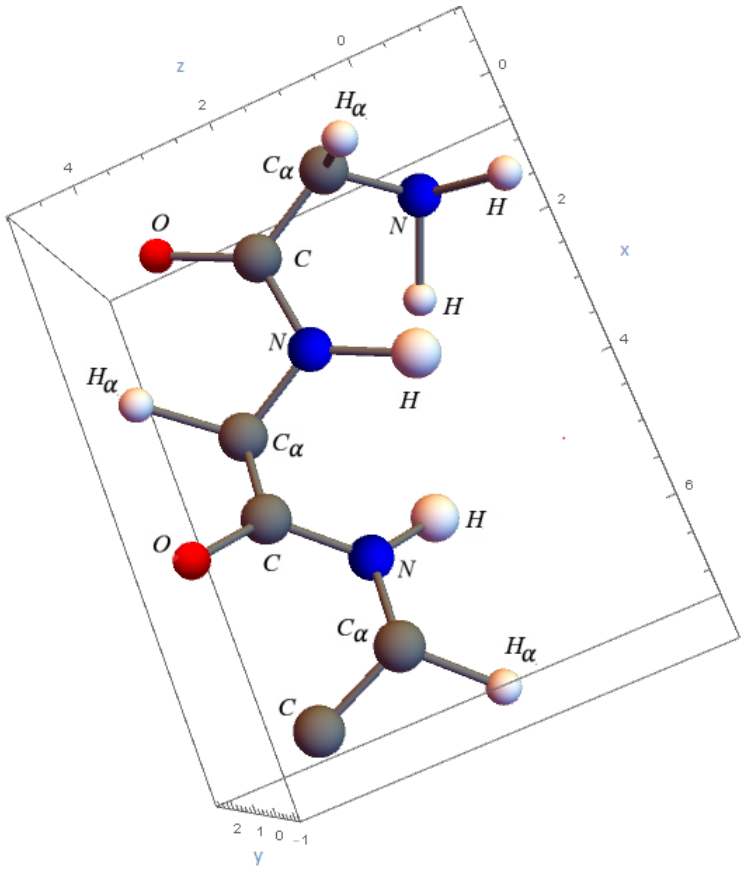

Figure 39 – Example of a representation of the main backbone of a protein containing three amino acids.

- Second tetrahedron $CNHC$ (figure 36)

$$\begin{aligned} x_{C_1} &= (0,0,0) \\ x_N &= (-r,0,0) \\ x_H &= (-r\cos\theta, -r\sin\theta, 0) \\ x_{C_2} &= p \end{aligned} \tag{6.7}$$

- Peptide plane $CNCOCH$ (figure 37)

$$\begin{aligned} C_1 &= (0,0,0) \\ N &= (-r,0,0) \\ C_2 &= (-r(1-\cos\theta), -r\sin\theta, 0) \\ O &= (-r\cos\beta, -r\sin\beta, 0) \\ C_3 &= (-r\cos\theta, r\sin\theta, 0) \\ H &= (-r(1-\cos\beta), r\sin\beta, 0) \end{aligned} \tag{6.8}$$

With the rigid substructures well defined, what is left is to find the torsion angle ω_{v_i} between them so that they can be correctly placed. More specifically, this is the angle between the two planes defined by $\{x_{v_{i-3}}, x_{v_{i-2}}, x_{v_{i-1}}\}$ and $\{x_{v_{i-2}}, x_{v_{i-1}}, x_{v_i}\}$, as shown in figure 40 for $i = 8$.

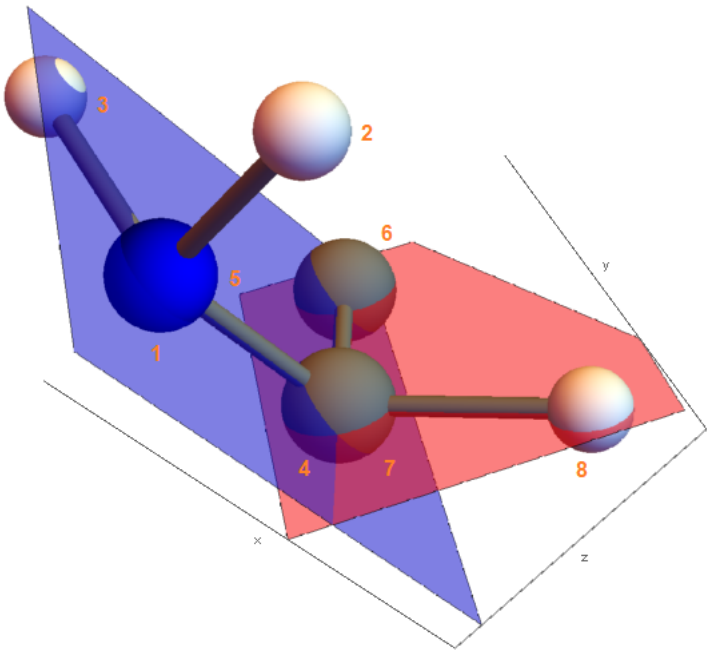

Figure 40 – Planes $x_{v_8}x_{v_7}x_{v_6}$ and $x_{v_7}x_{v_6}x_{v_5}$.

6.3 General Procedures

Using again the hand-crafted order to label the protein atoms it can be extracted a subset pertaining only to its hydrogens $\{h_1, h_2, ..., h_m\}$, where m is the number of hydrogens in the original order, and the general procedure of the algorithm will be to generate candidates for the realization h_w of the w^{th} hydrogen (the i^{th} atom in the original order). But instead of solving only one system like in the previous case, multiple systems having the form of system 6.2 will be solved. This happens due to the fact that the only information regarding distance $d_{v_i,v_{i-3}}$ is that it is one of the distances in the input list $\hat{d}$ and therefore all available distances must be tested. Each system will have the form

$$\begin{aligned} |x_{v_{i-1}} - x_{i_{\hat{d}_l}}| &= d_{v_{i,i-1}} \\ |x_{v_{i-2}} - x_{i_{\hat{d}_l}}| &= d_{v_{i,i-2}} \\ |x_{v_{i-3}} - x_{i_{\hat{d}_l}}| &= \hat{d}_l \end{aligned} \tag{6.9}$$

where $\hat{d}_l$ is the list of unused distances at that point of the algorithm and l varies between the elements of $\hat{d}_l$.

Since the previous atoms have a determined position and the molecule structure has rigid substructures, the placement of an hydrogen will always have a restrained circle

determined by the torsion angle of the current substructure being placed, as seen in figure 10.

For the second structure to be placed (that is, a second tetrahedron) the torsion angle will be around the axis that passes through atoms C and N (figure 41). For the third structure to be placed (peptide plane) and all following structures of this type, the torsion angle will be around the axis that passes through atoms C and C (figure 42). Finally, for the forth structure to be placed (that is, again a second tetrahedron) and all following structures of this type, the torsion angle will be around the axis that passes through atoms N and C (figure 43).

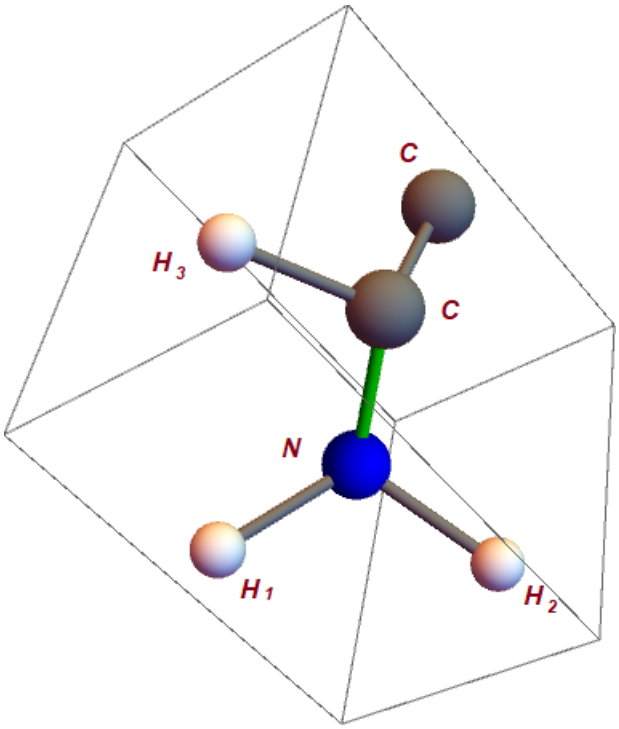

Figure 41 – Highlighted C-N axis (in green) regarding the rotation for the placement of atom H_3.

From equations 4.1 the coordinates for atom i are

$$\begin{matrix} -d_{v_{i-1},v_i} cos\theta_{v_{i-2},v_i} \\ d_{v_{i-1},v_i} sin\theta_{v_{i-2},v_i} cos\omega_{v_{i-3},v_i} \\ d_{v_{i-1},v_i} sin\theta_{v_{i-2},v_i} sin\omega_{v_{i-3},v_i} \end{matrix} \tag{6.10}$$

6.3.1 Prune

The pruning process is very similar to the prune performed in the classic BP algorithm described in the previous section, but there are a few differences. Firstly, if there is a cutoff value c and a distance calculated in a candidate is greater than c there is

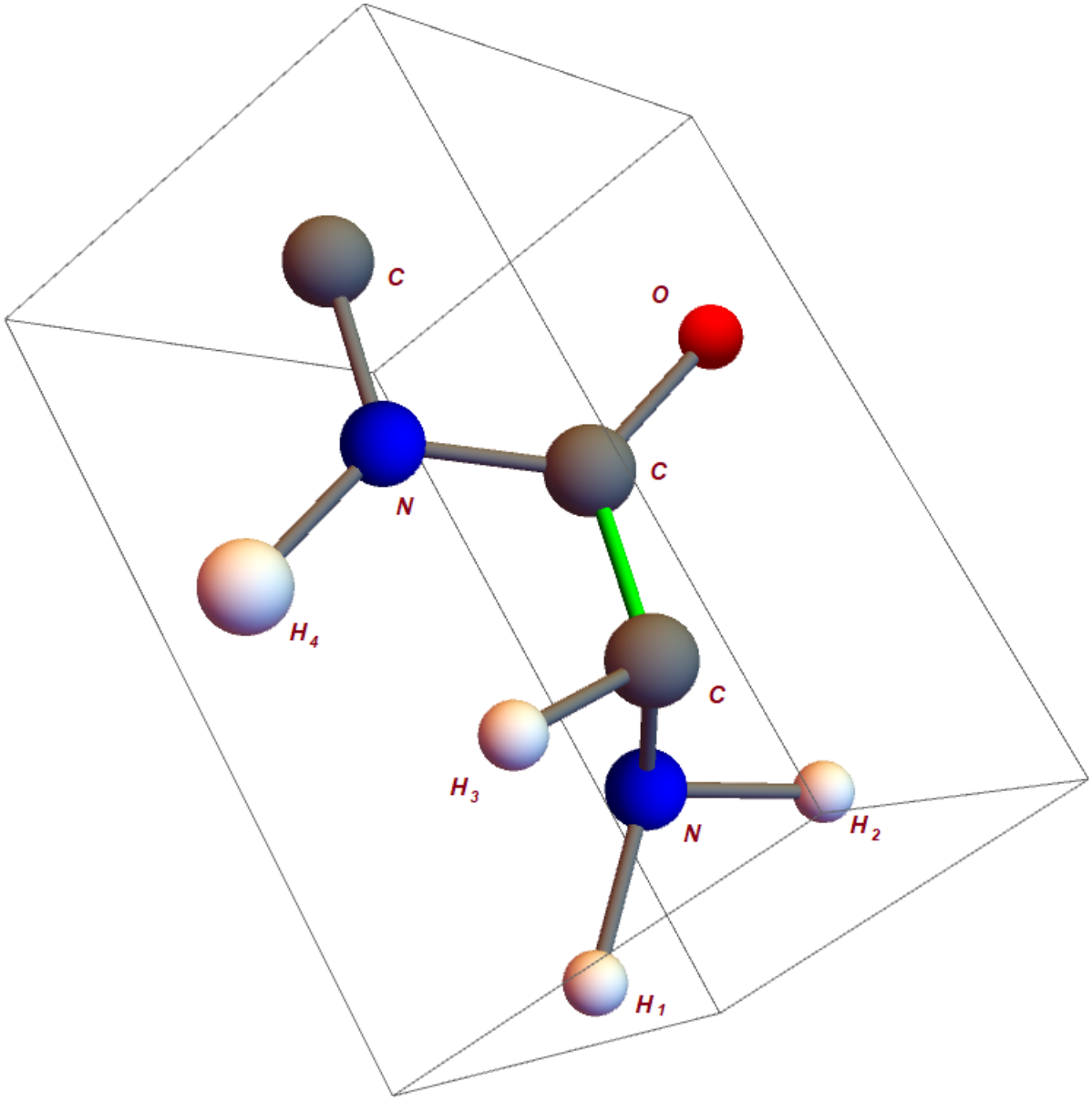

Figure 42 – Highlighted C-C axis (in green) regarding the rotation for the placement of atom H_4.

nothing to be done and the prune does not occur. If this calculated distance is less than or equal to c it can then be analyzed and a corresponding value can be looked for in the input data.

The number of distances associated to edges is closely monitored in the prune process. The reason for this is so that, if only a small number of them is associated and it is not possible to associate all distances by the last amino acid of that candidate, the candidate cannot be the correct one.

For example, consider an instance having 21 distances and three amino acids. If a candidate for the first two amino acids has only seven distances used, there is no way for all of the remaining 14 distances to be associated to pairs of atoms by the time the last

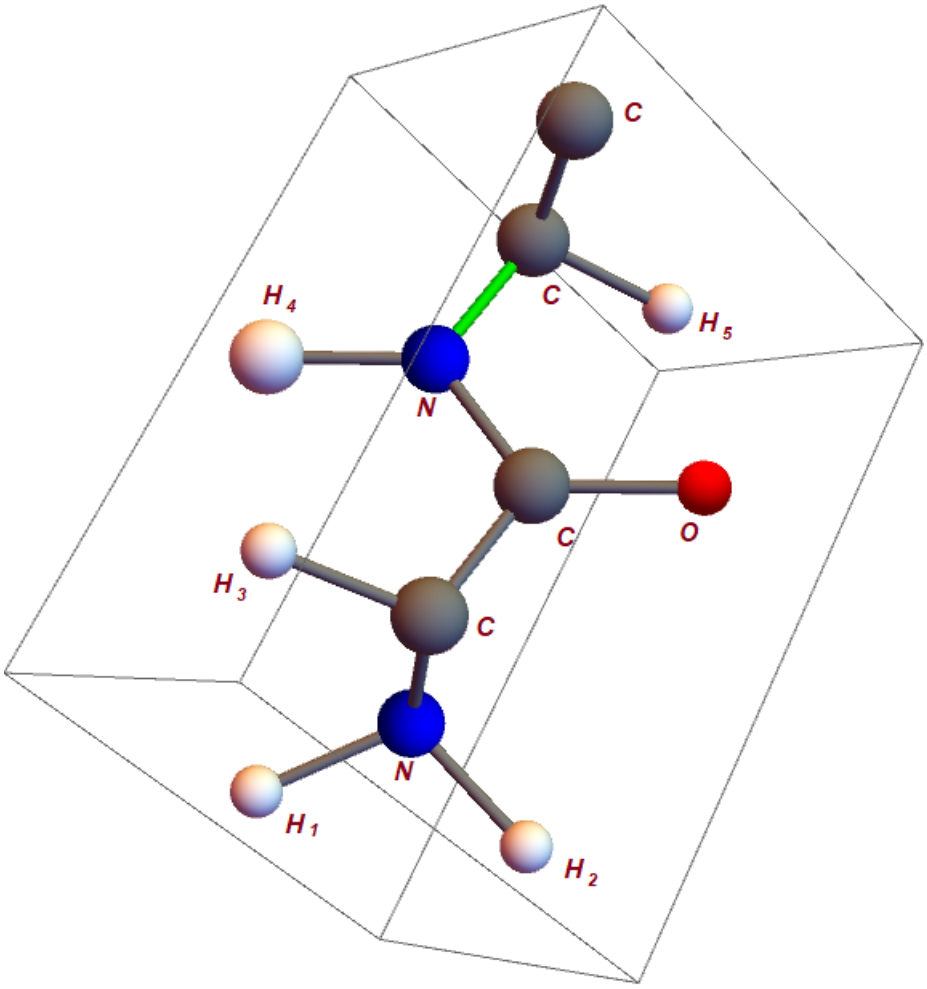

Figure 43 – Highlighted C-N axis (in green) regarding the rotation for the placement of atom H_5.

amino acid is placed in this structure. Hence, there is no point in continuing to explore this branch of the search tree of the algorithm, and this candidate can also be pruned. The pseudocode for the modified Branch-and-Prune algorithm is shown in algorithm 3.

6.4 Correctness of the Modified Branch-and-Prune Algorithm

Although the modified BP algorithm has been explained, it still needs to be certified that it works. This is called the correctness of the algorithm, which will be done in two parts: the first will assume that the algorithm ends and prove that it produces the correct answer. The second part will prove that it will always end.

Algorithm 3 Modified Branch-and-Prune Algorithm

Require: Number of amino acids p, cutoff value c and distance list $\hat{d}$
1: Set up rigid substructures
2: Priority line = [ ]
3: **while** A solution is not found **do**
4: Generate candidates from available distances
5: Prune
6: **if** There is at least one candidate **then**
7: Add corresponding rigid substructure to existing chain
8: Put last generated candidates at the front of priority line
9: Explore next candidate in priority line

There are some hypotheses here: since all instances are taken from an existing protein structure model, there will always be a solution. Also, the instances have a minimum number of available distances: for the realization of each hydrogen it is necessary that the distance of this hydrogen to the previous hydrogen in the order of the atoms of the protein is available.

It is known that the initial part of the structure (the amino acid composed of the first and second tetraheda) has three hydrogens (two in the first tetrahedron and one in the second) and after that each amino acid has two hydrogens (one from the second tetrahedron and one from the peptide plane). Thus, an instance containing p amino acids has $2p + 1$ hydrogens.

In order to realize the hydrogen i (where $i \geqslant 3$) at least one distance within the threshold of possible values has to be available. Therefore, given an instance with p amino acids it is necessary at least $2p$ distances. It will be assumed from now on that the input data agrees with this, as well as the fact that this distance list belongs to an existing structure.

In the first part of the proof there are three cases when realizing parts of a structure:

- The part of the structure trying to be realized is the first tetrahedron (figure 27)
- The part of the structure trying to be realized is the second tetrahedron (figure 28)
- The part of the structure trying to be realized is a peptide plane (figure 29)

The initial part of the demonstration regarding the realization of the first tetrahedron is trivial since it does not depend on the instance and therefore it will always be correct. Moreover, if H_1 and H_2 are defined as the two hydrogens in tetrahedron 1, the hydrogen H_3 will be the one in the next second tetrahedron and the hydrogen after, H_4, will be the one in the next peptide plane - as in figure 44.

Using this logic, for the following hydrogens $H_4, H_5, ..., H_{2p+1}$ (where p is the number of the molecule's amino acids), the hydrogens pertaining to the second tetrahedron will be regarded as H_{2i+1} and hydrogens pertaining to the peptide plane will be regarded as H_{2i}, for $i = 2, ..., p$.

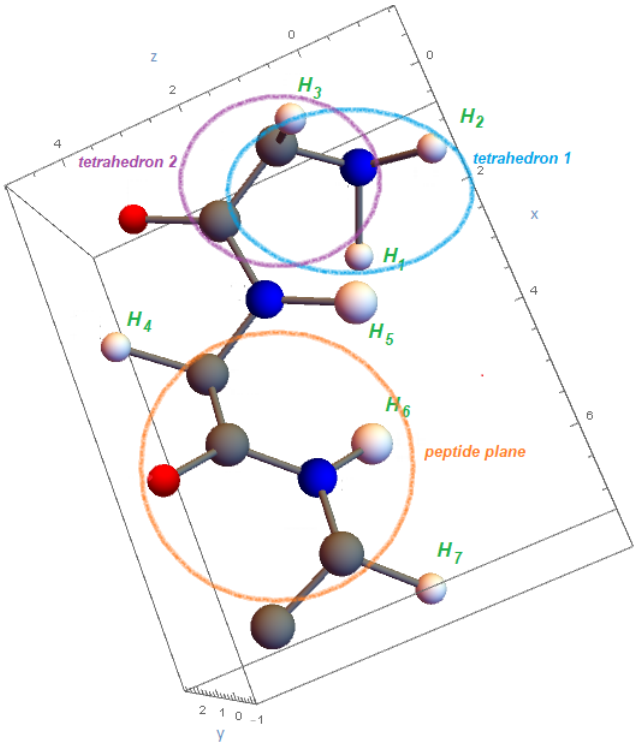

Figure 44 – Labeling of the hydrogens in a protein, associated with the rigid substructures they belong to.

The second tetrahedron and its following peptide plane are realized firstly by generating candidates for the position of the two hydrogens belonging to each of these structures, H_3 and H_4 respectively (as seen in figure 44). For this, as explained previously, distances not yet utilized from the input list are taken.

After generating all possible candidates it is necessary to fit the new structure in the amino acid chain, using the coordinates of the candidates as an anchor point. For the second tetrahedron, the other reference is the segment initiating in the Nitrogen and ending in the α-Carbon which is placed in the corresponding segment that already exists in the chain, and this plus the hydrogen coordinates makes the structure to be completely locked in.

Meanwhile, for the peptide plan, this part corresponds to the segment initiating in the α-Carbon and ending in the second Carbon of the previous tetrahedron. Since it is assumed that the instance originates from an existing structure it can be said that there is a distance in the available distances list that will generate the "correct" candidate.

Let it then be assumed that the algorithm is in its n^{th} step. The structure to be realized is either a peptide plane (when n is odd and greater than 1) or a second tetrahedron (when n is even). Assuming that this routine always ends, it will find the right partial solution (that is, the right structure up to the $n-1$ step) at some point, and since all possible available distances are used to generate the next candidate, one of these distances will generate the right one. Therefore, the first n parts of the structure are correct. Since n is generic, this argument can be applied to all iterations and therefore it is shown by induction that the right solution will be found.

In order to prove the second part one has to analyze the solution set of the problem. Since the algorithm walks through a search tree, the number of possible solutions is equal to the number of leafs this tree possesses. Each node that is not a leaf has a degree of $2|\hat{d}| + 1$ at most (where $|\hat{d}|$ is the cardinality of elements in the list given as the input). If the number of amino acids is p, the search tree will have a height of equal to $2p - 1$. Therefore, according to results from reference (WEST, 2001), it will have at most $(2|\hat{d}| + 1)^{(2p-1)}$ leafs. This shows that the search space is finite, and therefore the algorithm will finish.

6.5 Symmetries

As it was previously mentioned in chapter 5, the original Branch-and-Prune algorithm constructs a symmetric tree when used to find solutions for the Assigned Distance Geometry Problem's instances. Although a formal proof is yet to be provided, there is empiric evidence to support the fact that this is also the case with the Unassigned class. In figure 45 the search tree for the modified branch-and-prune algorithm, where the input of an instance was provided, is shown.

Figure 45 – Search tree of an uDGP intance.

The fact that the modified BP tests a greater number of distances than its original counterparts (with the "correct" distance among this set) implies that the solution set for the Unassigned class contains the solution set of the Assigned class. This conclusion, drawn from previous arguments and an original contribution in this work, can also be stated as:

Remark 1. *Let $\ddot{d}$ be an instance for the Assigned Distance Geometry Problem and $\ddot{d}_2$ the list having the same values but no vertex pairs (ie, its Unassigned counterpart). If S is the solution set for instance $\ddot{d}$ and S_2 is the solution set for instance $\ddot{d}_2$,*

$$S \subseteq S_2 \tag{6.11}$$

It can also be easily concluded that, if solutions for the aDGP are also solutions for the uDGP, their branches are also in the uDGP search tree and hence its properties are present:

Remark 2. *Remark 1 implies that the search tree of the Modified Branch-and-Prune Algorithm is partially symmetric*

These partial results added to empirical observations are evidence of the following:

Remark 3. *The search tree constructed by the Modified Branch-and-Prune Algorithm is (almost always) symmetric*

Research conducted about the Assigned Distance Geometry Problem (found in reference (LIBERTI et al., 2014b)) concluded that the cardinality of the solution set is a power of two. More specifically, it is equal to 2^{n-3}, where n is the number of atoms in the instance. Although there is evidence to support a similar result for the Unassigned Distance Geometry Problem, it is likely this quantity is greater than a power of two - possibly a power of a list of the size of the instance $|d_2|$.

The reason for this to be almost always the case rises from the nature of the problem when given a cutoff value, as shown, for instance, in figure 61. It can be the case, depending on the cutoff value c, that the calculated distance in the prune process is greater than c and less than c in the corresponding symmetric side of the tree. The distance smaller than c, when not found in the problem instance, results in the prune of one branch of the tree, while its symmetric counterpart cannot be pruned due to the absence of information for distances greater than c.

In the previous section an upper limit for the cardinality of the solution set was calculated as $(2|\hat{d}| + 1)^{(2p-1)}$, where p is the number of amino acids of the instance. The interval between this number and the size of the solution set for the Assigned problem is where $|S_2|$ lies, leading to a theorem originated from conclusions in this work:

Theorem 2. *Using the terminology stated in remark 1, the cardinality of the solution set S_2 is in the following interval:*

$$2^{n-3} \leqslant |S_2| \leqslant (2|\hat{d}| + 1)^{(2p-1)} \tag{6.12}$$

7 Computational Results

This chapter treats more practical aspects from this work (hence why points discussed in the sections ahead are grouped together). Firstly, the modified BP algorithm's procedures will be presented for examples of instances of the problem. Both instances in these examples have three amino acids - a small size but sufficient to illustrate the algorithm iterations.

Secondly, three instances derived from the same structure with different cutoff values are depicted and their search trees and solutions are analyzed. The goal of this example is to explore the possible symmetries that the search trees depict, and how different cutoff values have an effect in both the search trees and algorithm solutions.

Finally, computational results from tests who were run in instances having different combinations of sizes and cuts will be shown and commented, as well as suggestions for future actions regarding the subjects treated here.

7.1 Example 1

In this section it will be shown an example of the first iterations of the Modified Branch-and-Prune algorithm. The instance is derived from a structure with two amino acids and is given by

$$\hat{d} = [0.6972, 2.4913, 2.6464, 3.1882, 3.7331, 3.9363, 4.3073, 4.9666, 6.1670]$$

In this example, there is no cutoff value for the instance distances.

The algorithm begins calculating the first tetrahedron's coordinates, which are given in figure 46. The distance used to calculate this structure is equal to 2.4913 Å and is deleted from the available distances from this point on. For the next iteration the coordinates of the second tetrahedron will be calculated, starting from its hydrogen.

In order to find the coordinates for this hydrogen it is necessary to solve systems of the form

$$\begin{aligned} ||x_{H_3} - x_{N_1}|| &= d_{H_3,N_1} \\ ||x_{H_3} - x_{C_{\alpha 1}}|| &= d_{H_3,C_{\alpha 1}} \\ ||x_{H_3} - x_{H_2}|| &= \hat{d}_{H_{3,i},H_2} \end{aligned} \tag{7.1}$$

where $\hat{d}_{H_{3,i},H_2}$ is an unused distance from the input list (with i going from 1 to 10 since there are 10 possible distances to be used at this point). This means that, for every distance

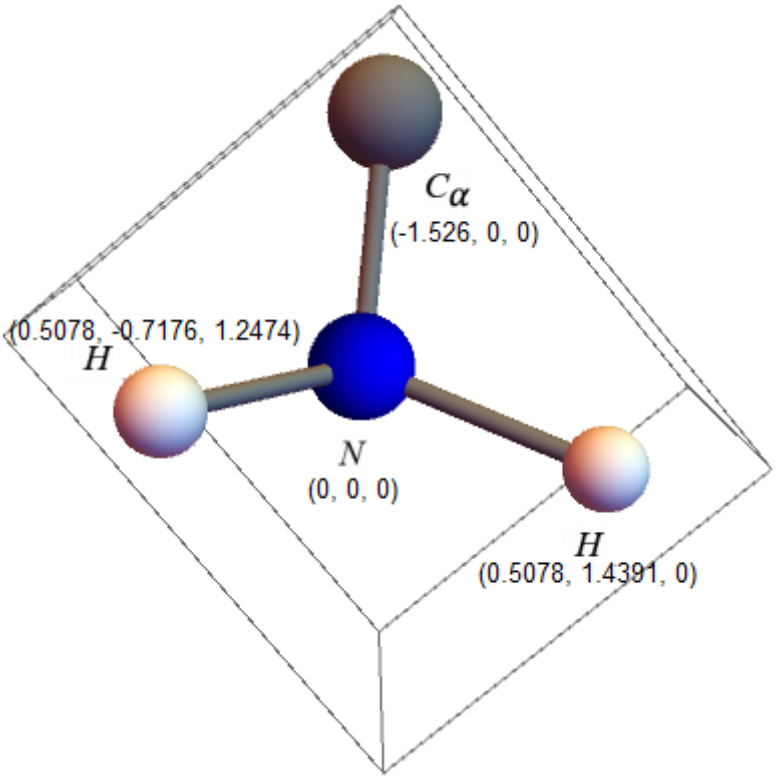

Figure 46 – Tetrahedron 1.

other than 2.4913 Å, a system of this form will be solved substituting the distance value. The candidates found are shown in table 1.

The next step is to eliminate candidates having complex coordinates. After

Distances	Coordinates
0.6972	no real solutions
2.6464	$(0.2779, 2.3169, 0.872697)$
	$(-1.0202, 2.2548, 0.2856)$
3.1882	$(0.6619, 1.5307, 1.8508)$
	$(-1.9434, 1.4062, 0.6726)$
3.2582	$(0.6307, 1.4145, 1.9514)$
	$(-1.9891, 1.2893, 0.7666)$
3.7331	$(-0.1858, 0.5305, 2.4271)$
	$(-1.7388, 0.4563, 1.7248)$
3.9363	no real solutions
4.3073	no real solutions
4.9666	no real solutions
6.1670	no real solutions

Table 1 – Candidates for iteration 2

Distances	Coordinates
2.6464	$(0.2779, 2.3169, 0.872697)$
	$(-1.0202, 2.2548, 0.2856)$
3.1882	$(0.6619, 1.5307, 1.8508)$
	$(-1.9434, 1.4062, 0.6726)$
3.2582	$(0.6307, 1.4145, 1.9514)$
	$(-1.9891, 1.2893, 0.7666)$
3.7331	$(-0.1858, 0.5305, 2.4271)$
	$(-1.7388, 0.4563, 1.7248)$

Table 2 – Candidates for iteration 2 having only real coordinates

that, the distances between H_{3i} and H_1 must be calculated and the prune process begins: if the calculated distances are not in the unused distances list, the associated structure can be discarded. The results are given in table 2.

Candidate	Calculated Distance	Is the Distance In the List
$(0.2779, 2.3169, 0.8726)$	3.2582	Yes
$(-1.0202, 2.2548, 0.2856)$	3.7648	No
$(0.6619, 1.5307, 1.8508)$	2.6922	No
$(-1.9434, 1.4062, 0.6726)$	3.7932	No
$(0.6307, 1.4145, 1.9514)$	2.6464	Yes
$(-1.9891, 1.2893, 0.7666)$	3.7661	No
$(-0.1858, 0.5305, 2.4271)$	2.6095	No
$(-1.7388, 0.4563, 1.7248)$	3.3265	No

Table 3 – Prune at the second iteration

And now there are two candidates left (as shown in table 1). For each one of them a second tetrahedron will be placed with the first tetrahedron using their coordinates as an anchor, forming two new structures. They will form the corresponding nodes at level two in the search tree, as shown in figure 47.

Taking one of these structures (shown in figure 47) and moving forward, it is necessary to generate the next candidates. The list of available distances at this point is $d_2 = [0.6972, 2.6464, 3.1882, 3.7331, 3.9363, 4.3073, 4.9666, 6.1670]$ and the algorithm moves forward in a similar fashion as it was shown in iteration 2.

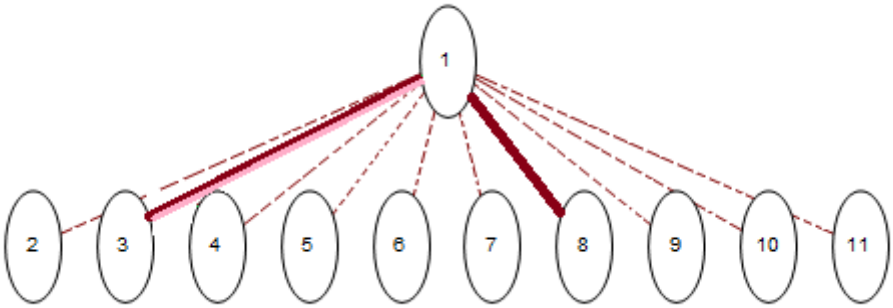

Figure 47 – Search tree at the end of iteration 2.

7.2 Example 2

In this second example an instance regarding a protein with three amino acids will be analyzed. This instance also has a cutoff value of 5.2 Å for all of its distances and is depicted below:

$$\hat{d} = [0.671034, 2.13928, 2.49139, 2.62566, 3.16099, 3.29366, \\ 3.77662, 3.88612, 3.76679, 4.24744, 4.70957, 5.10945]$$

As it was previously stated, the first thing to do is to calculate the coordinates for the first tetrahedron's coordinates, which are given in figure 46. The distance used to calculate this structure is equal to 2.4913 Å and is deleted from the available distances. For the next iteration the coordinates of the second tetrahedron will be calculated, starting from its hydrogen and using equations 7.1. The distance list for this iteration is given by

$$\hat{d}_2 = [0.671034, 2.13928, 2.62566, 3.16099, 3.29366, \\ 3.77662, 3.88612, 3.76679, 4.24744, 4.70957, 5.10945]$$

In order to find the coordinates for this hydrogen it is necessary to solve systems of the form

$$\begin{aligned} ||x_{H_3} - x_{N_1}|| &= d_{H_3,N_1} \\ ||x_{H_3} - x_{C_{\alpha 1}}|| &= d_{H_3,C_{\alpha 1}} \\ ||x_{H_3} - x_{H_2}|| &= \hat{d}_{H_{3,i},H_2} \end{aligned} \tag{7.2}$$

where $\hat{d}_{H_{3,i},H_2}$ is an unused distance from the input list (with i going from 1 to 9 since there are 9 possible distances to be used at this point). This means that, for every distance other than 2.4913, a system of this form will be solved substituting the distance value. The candidates found are shown in table 4, and the next thing to do is to exclude candidates with coordinates having imaginary numbers. These candidates and their respective distances are seen in table 5.

Distances	Coordinates
0.67103	no real solutions
2.13928	no real solutions
2.62566	$(0.222477, 2.34212, 0.81975)$
	$-0.946107, 2.28627, 0.291307)$
3.16099	$(0.669752, 1.57524, 1.81027)$
	$(-1.9217, 1.45139, 0.638392)$
3.29366	$(0.608639, 1.35442, 2.00053)$
	$(-2.00663, 1.22943, 0.817883)$
3.76679	$(-0.329331, 0.459582, 2.42639)$
	$(-1.63838, 0.39702, 1.83442)$
3.77662	$(-0.378128, 0.438376, 2.42319)$
	$(-1.60223, 0.379873, 1.86964)$
3.88612	no real solutions
4.24744	no real solutions
4.70957	no real solutions
5.10945	no real solutions

Table 4 – Candidates for iteration 2

Distances	Coordinates
2.62566	$(0.222477, 2.34212, 0.81975)$
	$-0.946107, 2.28627, 0.291307)$
3.16099	$(0.669752, 1.57524, 1.81027)$
	$(-1.9217, 1.45139, 0.638392)$
3.29366	$(0.608639, 1.35442, 2.00053)$
	$(-2.00663, 1.22943, 0.817883)$
3.76679	$(-0.329331, 0.459582, 2.42639)$
	$(-1.63838, 0.39702, 1.83442)$
3.77662	$(-0.378128, 0.438376, 2.42319)$
	$-1.60223, 0.379873, 1.86964)$

Table 5 – Candidates for iteration 2 (excluding complex solutions)

Going ahead, it is time to execute the pruning process. For the third iteration on, one way to look at the procedures is the following: the two candidates who were not pruned are put in a pile that prioritizes the last element to be placed, providing a way to span the tree of possibilities for the algorithm. Since newer elements are taken first and these elements are put in the order they are generated for the distances, the algorithm goes deep in the tree until a prune occurs or it reaches a leaf.

The highest candidate in the pile (coordinates $(0.608639, 1.35442, 2.00053)$) is taken for the next iteration. The previous list of available distances is upgraded according to this candidate, excluding from the list the distance that generated this candidate and the calculated distance to hydrogen 1. This makes this list take the form

Candidate	Calculated Distance	Is the Distance In the List
$(0.222477, 2.34212, 0.81975)$	3.29366	Yes
$(-0.514075, 2.40691, 0.386684)$	3.74843	No
$(-0.438819, 2.41725, 0.413977)$	3.62693	No
$(-1.55595, 1.91207, 0.360587)$	3.80207	No
$(-1.49154, 1.96672, 0.338142)$	2.62566	Yes
$(-0.148336, 2.42507, 0.551391)$	3.74985	No
$(0.144582, 2.37072, 0.752206)$	2.64919	No
$(-0.84655, 2.32335, 0.304009)$	3.25664	No
$(-1.36977, 2.05819, 0.307555)$	2.66414	No
$(-0.0167806, 2.40969, 0.63255)$	3.23301	No

Table 6 – Prune at the second iteration

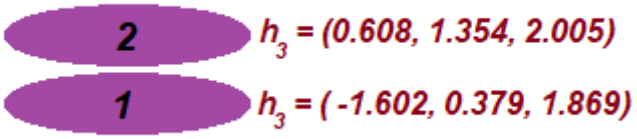

Figure 48 – Pile at the end of iteration 2 having coordinate candidates for the third hydrogen in the structure.

$[0.671034, 2.13928, 3.16099, 3.76679, 3.77662, 3.88612, 4.24744, 4.70957, 5.10945]$. These distances are each used to generate the next candidates that can be seen in table 7. Excluding complex solutions, the candidates are as shown in table 8.

And now the prune can be applied, with its results in table 9. From hereon now, when a distance can't be found in the distance list (meaning it can't be used to prune a candidate) the notation "−" will be used. Since this is the fourth hydrogen realized (using, among other elements, the coordinates of the third hydrogen), there are two available hydrogens (hydrogen 1 and hydrogen 2) to be used in order to compare the distances from them to these candidates and check them against the list of available distances. Hence, there are two distances displayed in the column for the calculated distances and two checks are made for whether they are in the distance list or not.

A solution candidate is not pruned only if the following requirements are satisfied: both calculated distances are in the list, if this distance pair has one distance being larger than the cutoff value (what would result in these distances being excluding

Distances	Coordinates
0.67103	no real solutions
2.13928	$(-0.3679, 1.4683, 3.9004)$
	$(0.1671, 0.1808, 3.7338)$
3.16099	$(-2.0328, 3.0148, 2.5082)$
	$(-0.0594, -1.7333, 1.8936)$
3.76679	$(-2.7615, 2.8623, 1.2543)$
	$(-0.7965, -1.8657, 0.6423)$
3.77662	$(-2.7716, 2.8528, 1.2313)$
	$(-0.8124, -1.8612, 0.6211)$
3.88612	$(-2.8775, 2.7266, 0.9682)$
	$(-0.9995, -1.7920, 0.3833)$
4.24744	$(-3.0953, 1.9162, -0.0008)$
	$(-1.8081, -1.1811, -0.4017)$
4.70957	no real solutions
5.10945	no real solutions

Table 7 – Candidates for iteration 3

Distances	Coordinates
2.13928	$(-0.3679, 1.4683, 3.9004)$
	$(0.1671, 0.1808, 3.7338)$
3.16099	$(-2.0328, 3.0148, 2.5082)$
	$(-0.0594, -1.7333, 1.8936)$
3.76679	$(-2.7615, 2.8623, 1.2543)$
	$(-0.7965, -1.8657, 0.6423)$
3.77662	$(-2.7716, 2.8528, 1.2313)$
	$(-0.8124, -1.8612, 0.6211)$
3.88612	$(-2.8775, 2.7266, 0.9682)$
	$(-0.9995, -1.7920, 0.3833)$
4.24744	$(-3.0953, 1.9162, -0.0008)$
	$(-1.8081, -1.1811, -0.4017)$

Table 8 – Candidates for iteration 3 (excluding complex solutions)

from the comparison) and the distance smaller than the cutoff value (if this distance exists) is in the list or if both calculated distances are larger than the cutoff value. For this iteration all of the calculated distances being below the cutoff value of 5.2 Å are compared against the list of available distances.

Since all of the candidates have been pruned, the algorithm goes ahead on the pile (figure 50) and tests the other candidate for the third hydrogen: the coordinates $(0.2224, 2.3421, 0.8197)$. At this point the list of available distances also changes, going back to the beginning of iteration 2 with the modification that the distance that generated this candidate (equal to 2.62566) and the calculated distance of the prune (equal to 3.29366)

Candidate	Calculated Distances (hydrogen 1, hydrogen 2)	Are the Distances In the List
$(-0.3679, 1.4683, 3.9004)$	$\{4.26895, 5.26155\}$	{No, − }
$(0.1671, 0.1808, 3.7338)$	$\{3.40778, 5.00964\}$	{No, No}
$(-2.0328, 3.0148, 2.5082)$	$\{5.31494, 5.07284\}$	{ − , No}
$(-0.0594, -1.7333, 1.8936)$	$\{1.96697, 4.02388\}$	{No, No}
$(-2.7615, 2.8623, 1.2543)$	$\{5.39268, 4.63759\}$	{ − , No}
$(-0.7965, -1.8657, 0.6423)$	$\{2.19184, 3.46488\}$	{No, No}
$(-2.7716, 2.8528, 1.2313)$	$\{5.3908, 4.62805\}$	{ − , No}
$(-0.8124, -1.8612, 0.6211)$	$\{2.20355, 3.45617\}$	{No, No}
$(-2.8775, 2.7266, 0.9682)$	$\{5.36058, 4.51459\}$	{ − , No}
$(-0.9995, -1.7920, 0.3833)$	$\{2.35243, 3.36162\}$	{No, No}
$(-3.0953, 1.9162, -0.0008)$	$\{5.07006, 4.00098\}$	{No, No}
$(-1.8081, -1.1811, -0.4017)$	$\{3.13079, 3.127827\}$	{No, No}

Table 9 – Prune for iteration 3

are removed. This distance list is now equal to

$$[0.671034, 2.13928, 3.16099, 3.76679, 3.77662, 3.88612, 4.24744, 4.70957, 5.10945] \quad (7.3)$$

and the corresponding partial structure at this point is shown in figure 49.

Using the available distances, the next candidates are generated as seen in table 10. In table 11 the remaining candidates after the exclusion of imaginary numbers are shown. The next step is to test these candidates in the pruning process: having two candidates that have not been pruned, the algorithm stores them again in a pile and proceeds to extract the highest candidate, in this case the one related to the coordinate $(1.3170, -1.2625, 1.08709)$. The distance list now is equal to $[2.13928, 3.76679, 3.88612, 4.24744, 4.70957, 5.1094]$, the related candidates generated by them are in table 13 and the candidates having only real numbers are shown in table 14. The corresponding partial structure at this point is shown in figure 51.

The prune process now compares the distances between the candidates and

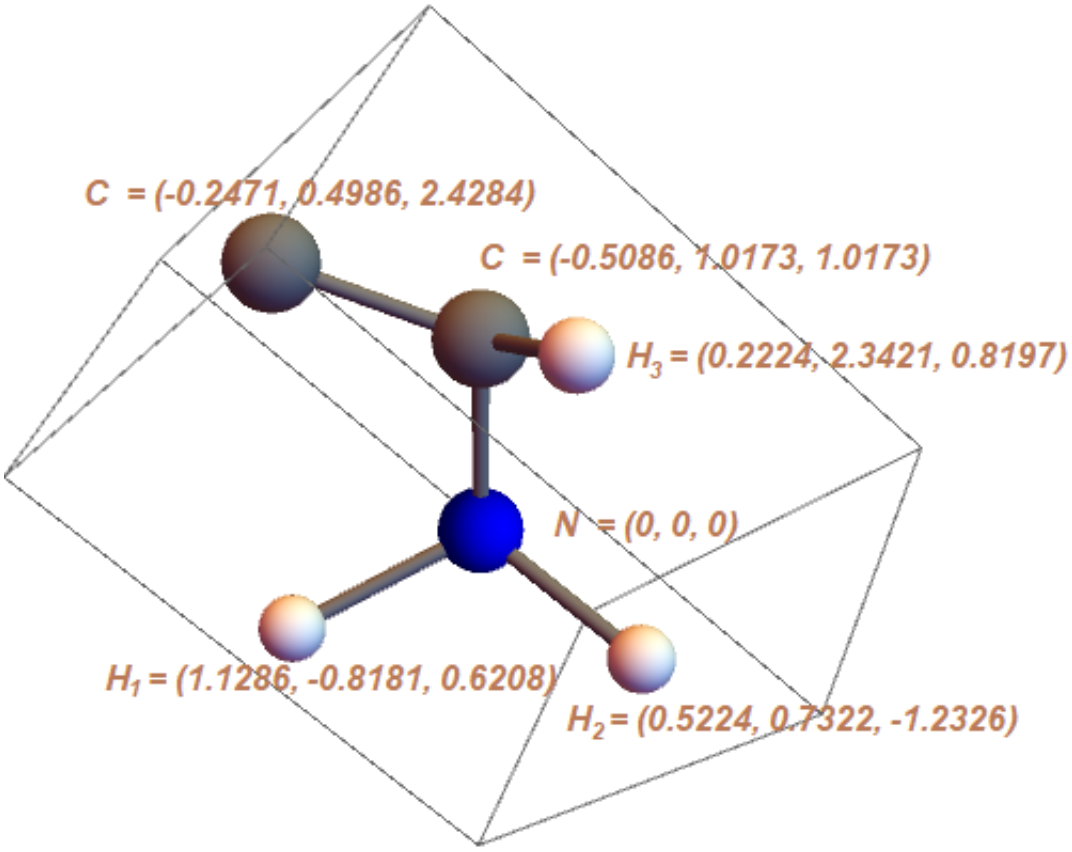

Figure 49 – Partial structure at the beginning of iteration 3.

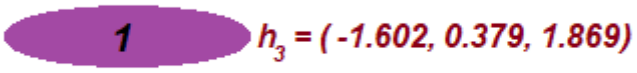

Figure 50 – Pile at the end of iteration 3 having one last coordinate candidate for the third hydrogen in the structure.

hydrogens 1, 2 and 3, with its results shown in table 15. Only one candidate (the one having its coordinates equal to $(-0.2853, -1.0350, 4.6201)$) is not pruned, so for the next iteration this candidate will be regarded as the fifth hydrogen in the structure. The candidate for the fourth hydrogen shown is position 1 in the pile of figure 52 will remain there waiting to be checked. The corresponding partial structure at this point is shown in figure 53.

Distances	Coordinates
0.67103	no real solutions
2.13928	$(0.5971, 3.0467, 2.8046)$
	$(1.7269, 2.354, 2.3405)$
3.16099	$(-1.9910, 2.4712, 3.0726)$
	$(2.1756, -0.0835, 1.3613)$
3.76679	$(-2.8112, 1.2983, 2.7934)$
	$(1.3378, -1.2456, 1.0894)$
3.77662	$(-2.8197, 1.2739, 2.7860)$
	$(1.3170, -1.2625, 1.08709)$
3.88612	$(-2.8998, 0.9879, 2.6957)$
	$(1.0655, -1.4433, 1.0672)$
4.24744	$(-2.8566, -0.2119, 2.2467)$
	$(-0.1386, -1.8785, 1.1303)$
4.70957	no real solutions
5.10945	no real solutions

Table 10 – Candidates for iteration 4

Distances	Coordinates
2.13928	$(0.5971, 3.0467, 2.8046)$
	$(1.7269, 2.354, 2.3405)$
3.16099	$(-1.9910, 2.4712, 3.0726)$
	$(2.1756, -0.0835, 1.3613)$
3.76679	$(-2.8112, 1.2983, 2.7934)$
	$(1.3378, -1.2456, 1.0894)$
3.77662	$(-2.8197, 1.2739, 2.7860)$
	$(1.3170, -1.2625, 1.08709)$
3.88612	$(-2.8998, 0.9879, 2.6957)$
	$(1.0655, -1.4433, 1.0672)$
4.24744	$(-2.8566, -0.2119, 2.2467)$
	$(-0.1386, -1.8785, 1.1303)$

Table 11 – Candidates for iteration 4 (excluding complex solutions)

For the next iteration, the updated list of available distances is

$$\hat{d}_6 = [2.13928, 3.76679, 4.70957]$$

and the generated candidates are shown in table 16 and the subset of candidates having no imaginary numbers is shown in table 17. The corresponding partial structure at this point is shown in figure 53. In this instance of the prune, there are two candidates that have all of their calculated distances larger than the cutoff value. Hence, there is no way to compare and prune them and so they both are considered as candidates for the

Candidate	Calculated Distances (hydrogen 1, hydrogen 2)	Are the Distances In the List
$(0.5971, 3.0467, 2.8046)$	$\{4.47092, 4.65427\}$	{No, − }
$(1.7269, 2.354, 2.3405)$	$\{3.65762, 4.10476\}$	{No, No}
$(-1.9910, 2.4712, 3.0726)$	$\{5.15403, 5.27992\}$	{ − , No}
$(2.1756, -0.0835, 1.3613)$	$\{1.47796, 3.18242\}$	{No, No}
$(-2.8112, 1.2983, 2.7934)$	$\{4.97221, 5.25777\}$	{ No, − }
$(1.3378, -1.2456, 1.0894)$	$\{0.66786, 3.15741\}$	{No, No}
$(-2.8197, 1.2739, 2.7860)$	$\{4.96541, 5.25495\}$	{ Yes, −}
$(1.3170, -1.2625, 1.08709)$	$\{0.67103, 3.161\}$	{Yes, Yes}
$(-2.8998, 0.9879, 2.6957)$	$\{4.8781, 5.21633\}$	{ No, −}
$(1.0655, -1.4433, 1.0672)$	$\{0.770748, 3.21215\}$	{No, No}
$(-2.8566, -0.2119, 2.2467)$	$\{4.34667, 4.94127\}$	{No, No}
$(-0.1386, -1.8785, 1.1303)$	$\{1.72912, 3.58293\}$	{No, No}

Table 12 – Prune for iteration 4

Distances	Coordinates
2.13928	no real solutions
3.76679	$(1.4704, -0.0596, 4.6533)$
	$(-0.3550, -1.5277, 4.4519)$
3.88612	$(0.95614, -0.03670, 4.7571)$
	$(-0.2853, -1.0350, 4.6201)$
4.24744	no real solutions
4.70957	no real solutions
5.10945	no real solutions

Table 13 – Candidates for iteration 5

Distances	Coordinates
3.76679	$(1.4704, -0.0596, 4.6533)$
	$(-0.3550, -1.5277, 4.4519)$
3.88612	$(0.95614, -0.03670, 4.7571)$
	$(-0.2853, -1.0350, 4.6201)$

Table 14 – Candidates for iteration 5 (excluding complex solutions)

Candidate	Calculated Distances (hydrogen 1, hydrogen 2, hydrogen 3)	Are the Distances In the List
$(1.4704, -0.0596, 4.6533)$	$\{4.11744, 6.01428$ $4.69284\}$	{No, −, No }
$(-0.3550, -1.5277, 4.4519)$	$\{4.1692, 6.18005,$ $5.33875\}$	{No, −, −}
$(0.95614, -0.03670, 4.7571)$	$\{4.21297, 6.05452,$ $4.65832\}$	{No, −, No }
$(-0.2853, -1.0350, 4.6201)$	$\{4.24744, 6.167,$ $5.10945\}$	{Yes, −, Yes}

Table 15 – Prune for iteration 5

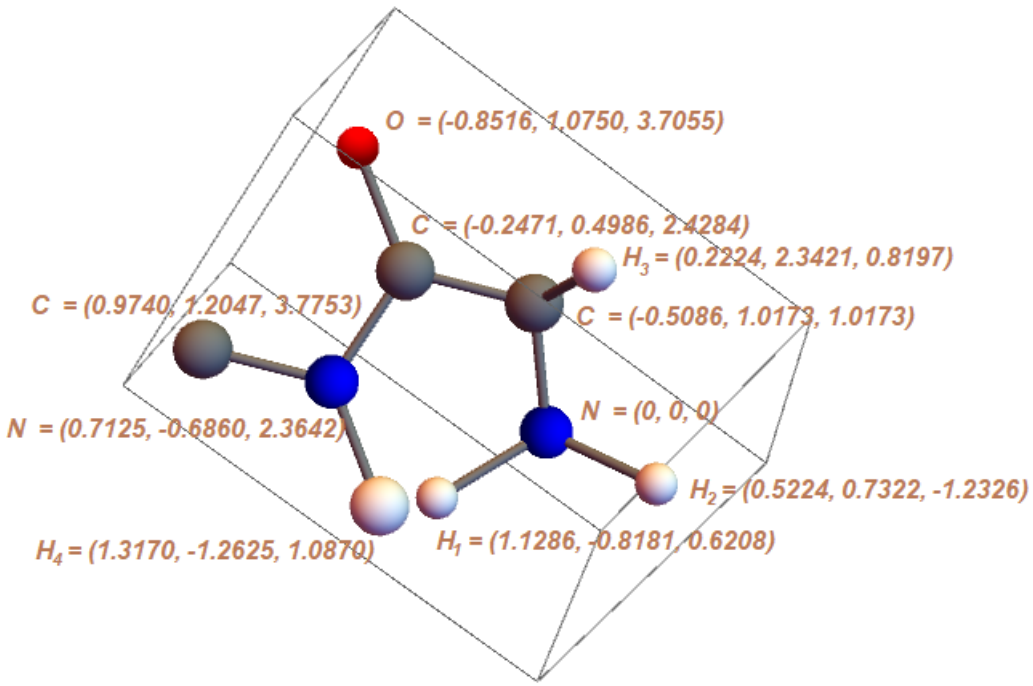

Figure 51 – Partial structure at the beginning of iteration 4.

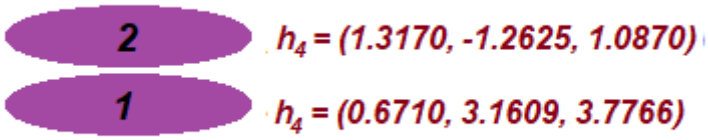

Figure 52 – Pile at the end of iteration 4 having two coordinate candidates for the fourth hydrogen in the structure.

next iteration. One of them is taken for the next step of the algorithm while the other is placed into the pile previously mentioned, that now has the configuration as shown in figure 55.

The next iteration generates only one candidate after the pruning process, and since this is the last part of the protein structure, the first solution has been found. The algorithm now goes back to the same configuration it had at the beginning of iteration 6 with the addition of the peptide plane containing the remaining candidate at the pile. The distance list is also modified to the distance list at iteration 7 minus the distance that

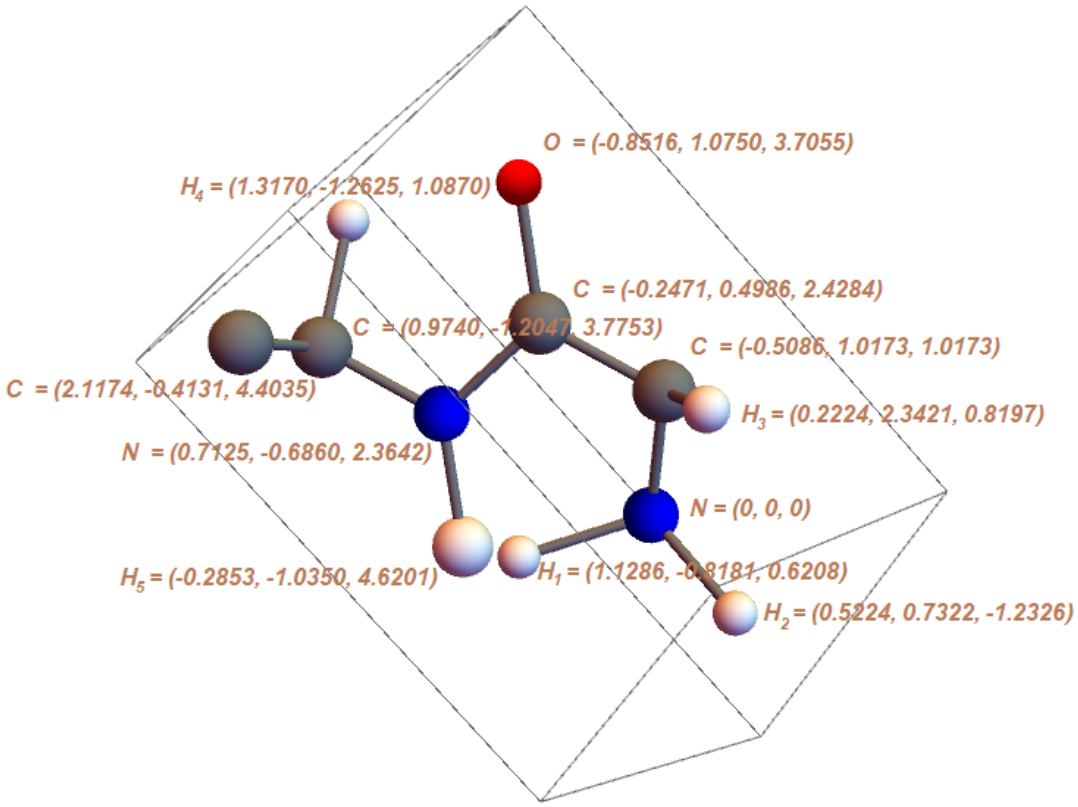

Figure 53 – Partial structure at the beginning of iteration 5.

Distances	Coordinates
2.13928	$(0.4344, -0.3888, 6.5282)$
	$(0.0749, 0.7347, 5.7666)$
3.76679	$(2.9546, -2.8953, 5.0999)$
	$(1.6345, 1.2308, 2.3031)$
4.70957	no real solutions

Table 16 – Candidates for iteration 6

generated this candidate and, potentially, distances paired to this candidate in the pruning process (which in these case are none since all calculated distances for this candidate were larger than the cutoff value, so only the distance that generated this candidate is removed).

At iteration 8, after the candidates are generated (table 22) and those having only real coordinates are filtered (table 23), the prune is performed and only one candidate remains not pruned (table 24). Since this is the last hydrogen to be added in the structure, another solution has been found. Without other candidates left to analyze, the algorithm terminates at this point. The corresponding final solutions are shown in figures 56 and 57.

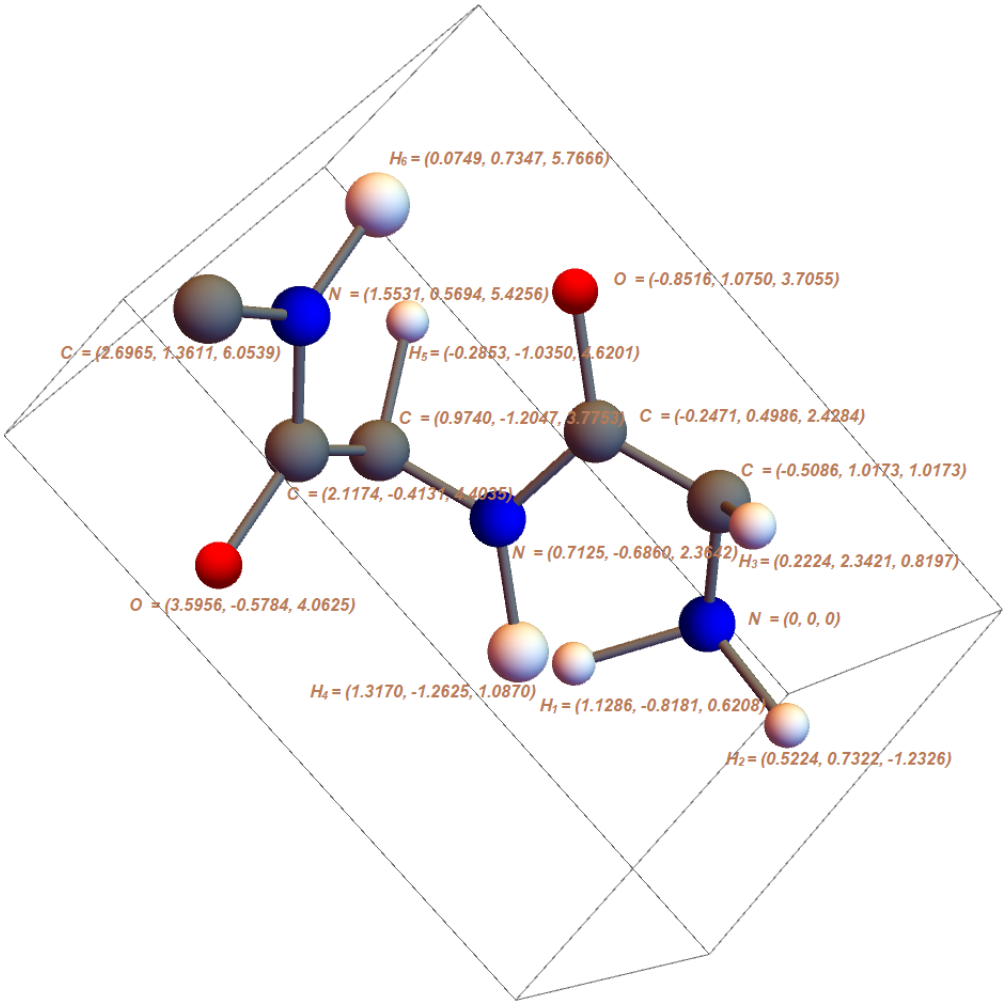

Figure 54 – Partial structure at the beginning of iteration 6.

Distances	Coordinates
2.13928	$(0.4344, -0.3888, 6.5282)$
	$(0.0749, 0.7347, 5.7666)$
3.76679	$(2.9546, -2.8953, 5.0999)$
	$(1.6345, 1.2308, 2.3031)$

Table 17 – Candidates for iteration 6 (excluding complex solutions)

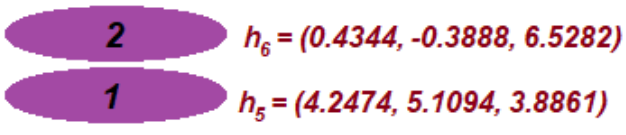

Figure 55 – Pile at the start of iteration 7 having two coordinate candidates for the fourth hydrogen in the structure.

Candidate	Calculated Distances (hydrogen 1, hydrogen 2, hydrogen 3, hydrogen 4)	Are the Distances In the List
$(0.4344, -0.3888, 6.5282)$	$\{5.96353, 7.84202,$ $6.33171, 5.5811\}$	{–, –, –, –}
$(2.9546, -2.8953, 5.0999)$	$\{5.26419, 7.69274,$ $7.29497, 4.63157\}$	{–, –, –, No}
$(0.0749, 0.7347, 5.7666)$	$\{5.47733, 7.01364,$ $5.20359, 5.23739\}$	{–, –, –, –}
$(1.6345, 1.2308, 2.3031)$	$2.69902, 3.74001,$ $2.33009, 2.79223\}$	{No, No, No, No}

Table 18 – Prune for iteration 6

Distances	Coordinates
3.76679	$(3.7245, 0.3943, 6.6345)$
	$(3.5191, 2.0251, 4.9534)$
4.70957	no real solutions

Table 19 – Candidates for iteration 7

Distances	Coordinates
3.76679	$(3.7245, 0.3943, 6.6345)$
	$(3.5191, 2.0251, 4.9534)$

Table 20 – Candidates for iteration 7 (excluding complex solutions)

Candidate	Calculated Distances (hydrogen 1, hydrogen 2, hydrogen 3, hydrogen 4, hydrogen 5)	Are the Distances In the List
$(3.7245, 0.3943, 6.6345)$	$\{6.66137, 8.50069,$ $7.06192, 6.27024,$ $4.70958\}$	{–, –, –, –, Yes }
$(3.5191, 2.0251, 4.9534)$	$6.33171, 5.5811\}$ $\{5.70703, 6.99426,$ $5.29675, 5.53231,$ $4.89385\}$	{–, –, –, –, No}

Table 21 – Prune for iteration 7

Distances	Coordinates
3.76679	$(4.1816, -0.0456, 6.3570)$
	$(3.2250, 1.9553, 5.5765)$
4.70957	no real solutions

Table 22 – Candidates for iteration 8

Distances	Coordinates
3.76679	$(4.1816, -0.0456, 6.3570)$
	$(3.2250, 1.9553, 5.5765)$

Table 23 – Candidates for iteration 8 (excluding complex solutions)

Candidate	Calculated Distances (hydrogen 1, hydrogen 2, hydrogen 3, hydrogen 4, hydrogen 5)	Are the Distances In the List
$(4.1816, -0.0456, 6.3570)$	$\{6.54384, 8.46165, 7.21377, 6.12041, 4.89385\}$	{−, −, −, −, No }
$(3.2250, 1.9553, 5.5765)$	$\{6.05361, 7.42736, 5.63847, 5.84381, 4.70958\}$	{−, −, −, −, Yes}

Table 24 – Prune for iteration 8

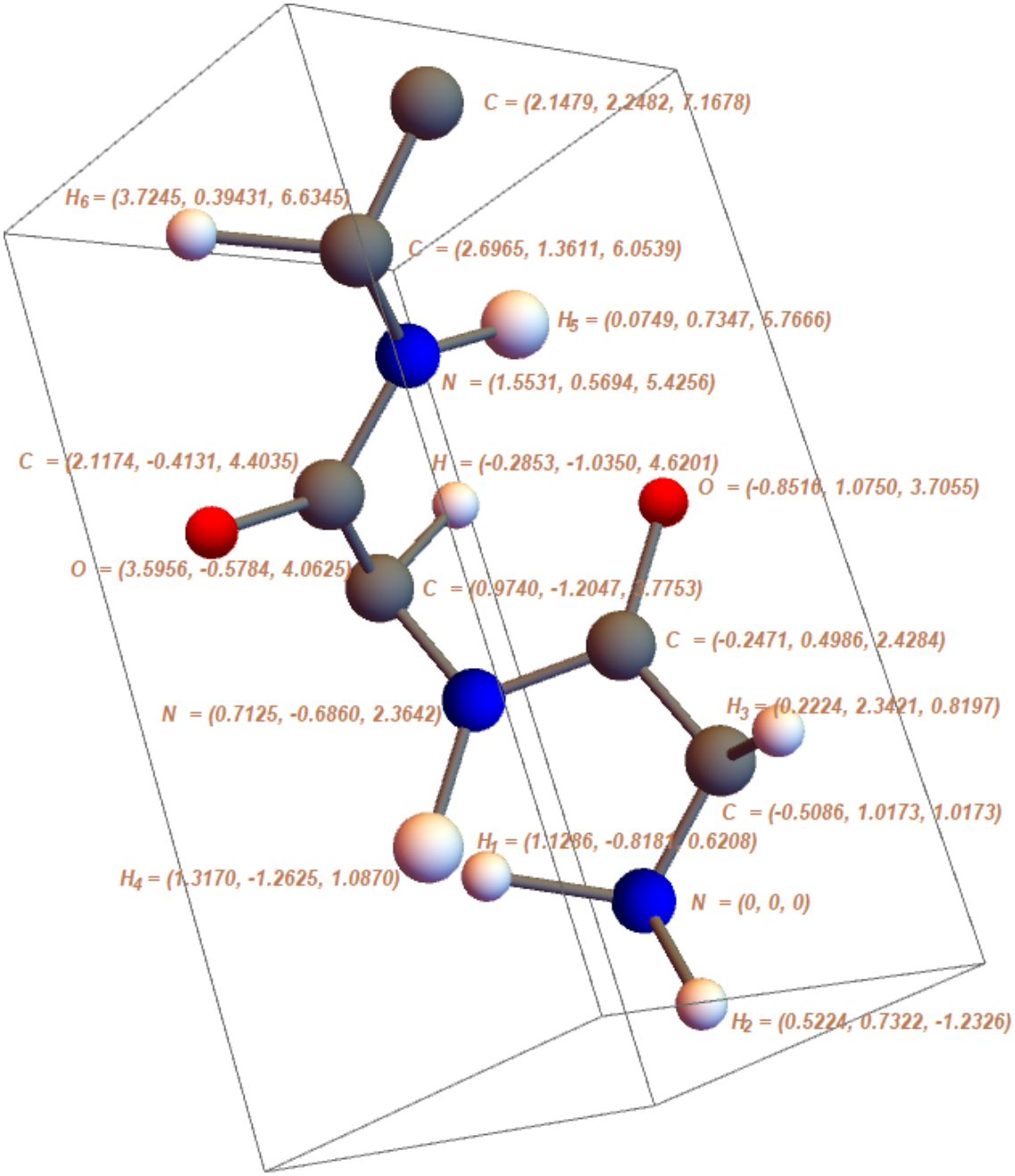

Figure 56 – First solution.

7.3 Analysis of Different Cutoff Values

One of the most important results related to the Branch-and-Prune algorithm for the Assigned Distance Geometry Problem, found in reference (MUCHERINO C. LAVOR, 2012), is about the symmetric characteristic of its search tree. With this in mind, it is easy to realize the importance of analyzing possible symmetries for the Unassigned case.

The idea here is to use a single structure (shown in figure 58) as an example

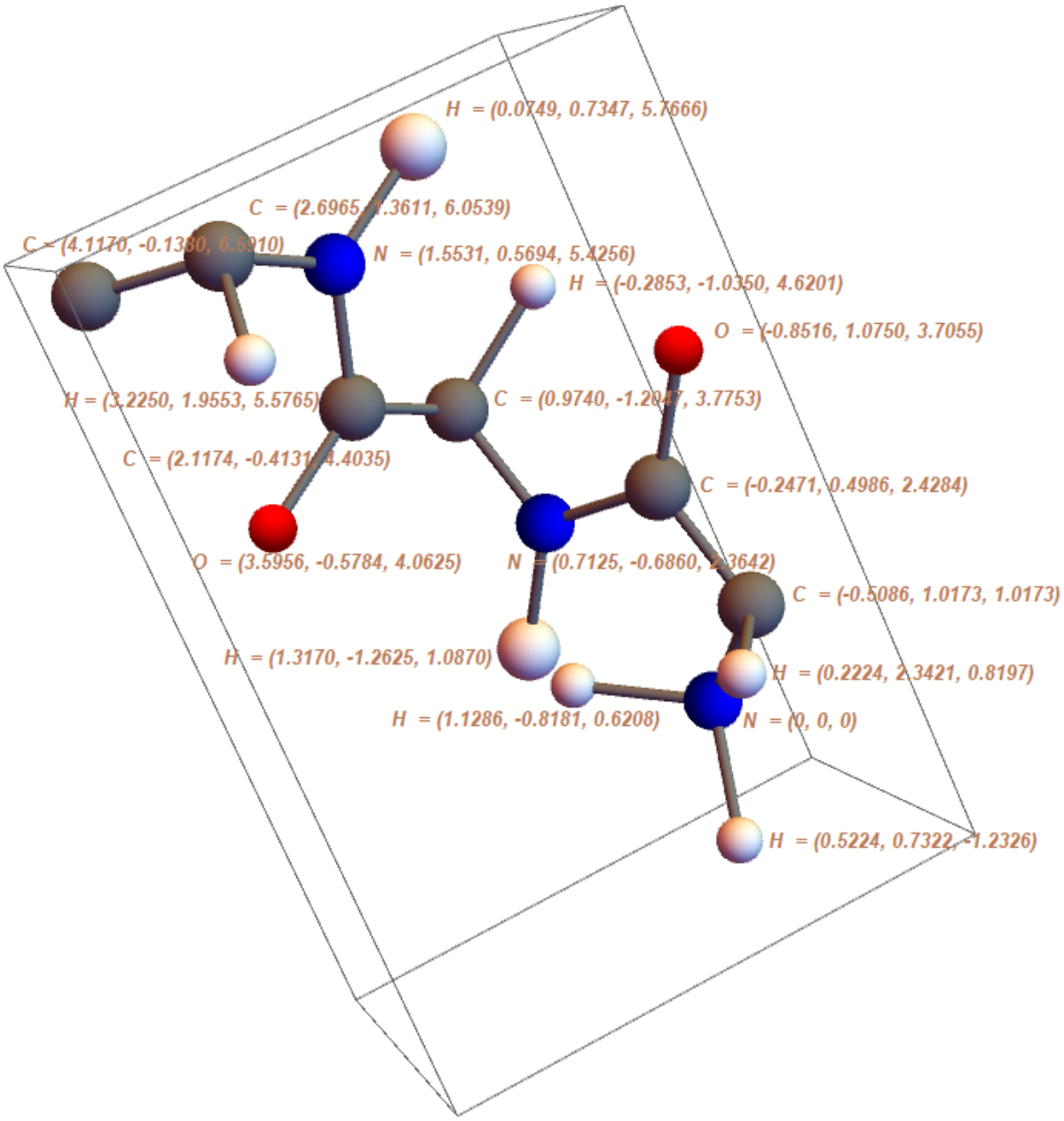

Figure 57 – Second solution.

for the analysis and take a closer look at the search tree generated by the BP algorithm when given the instances generated by this structure as input data. This process is to be repeated with different cutoff values so that their impact on the number of solutions and on the format of the search tree can be verified.

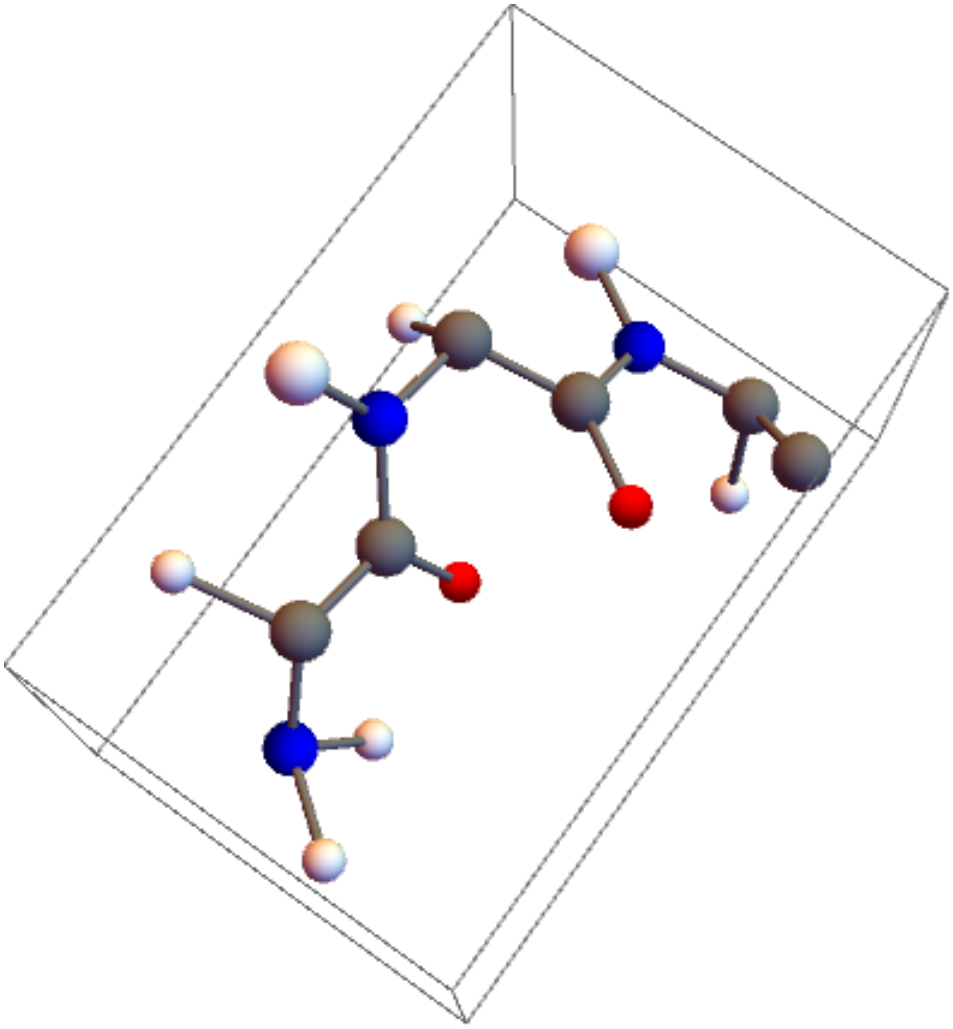

Figure 58 – Real solution that generates the instances for the symmetry analysis.

The cutoff values shown ahead will be listed in increasing order. Starting with the smallest value shows a fuller search tree and, consequently, a higher number of solutions. This happens due to the fact that a smaller cutoff value implies a smaller number of distances available in the instance and also a smaller number of viable prunes, since the chance of creating a candidate with distances to the other hydrogens of the structure being over this cutoff value increases as the cutoff value gets smaller.

For this example three cutoff values will be considered: 4.5 Å, 4.6 Å and 4.9 Å. With the original structure (figure 58) in hands, the distances between its hydrogens are calculated and the result is the following:

2.49139	2.53149	2.65238	3.43621	3.52491	3.59369	3.6266
4.58485	4.62605	4.91384	5.09236	5.28422	5.4983	5.58578
6.47463	6.59986	6.91666	7.00581	7.39289	7.576	8.30219

The next step is to filter this distance list according to each cutoff value, selecting all values less than or equal to these cutoff values and generating the instances for each test. The result is shown ahead:

$$\text{CUTOFF EQUAL TO 4.5 Å:} \begin{cases} 2.49139 & 2.53149 & 2.65238 \\ 3.43621 & 3.52491 & 3.59369 \\ 3.6266 & & \end{cases}$$

$$\text{CUTOFF EQUAL TO 4.6 Å:} \begin{cases} 2.49139 & 2.53149 & 2.65238 \\ 3.43621 & 3.52491 & 3.59369 \\ 3.6266 & 4.58485 & \end{cases}$$

$$\text{CUTOFF EQUAL TO 4.9 Å:} \begin{cases} 2.49139 & 2.53149 & 2.65238 \\ 3.43621 & 3.52491 & 3.59369 \\ 3.6266 & 4.58485 & 4.62605 \end{cases}$$

In figures 59, 60 and 61 the search trees for each analyzed cutoff value are depicted. With the addition of distances as the cutoff values are increased no new branch is created since the added distances did not result in strict real number coordinates for these candidates. Instead, these distances were used in the prune part of the algorithm, selecting or pruning candidates when some extra calculated distance in the prune is less than or equal to the cutoff value. Hence, when this threshold increases the search tree diminishes.

From these examples it is possible to see some symmetry: both sides of the tree are symmetric (or almost symmetric in the case of the cutoff value equal to 4.6 Å). As the cutoff values change, so does the search tree - but these changes are also almost always symmetric. When they are not, as it happens in the case of the search tree in figure 61

where the node 48 is pruned but the node 106 is not, it usually happens due to the extra calculated distances.

Indeed, it can happen that, with the increase of the cutoff value, some calculated distances in the prune operation can be checked when they could not have been before. In the particular case of figure 61, at least one of the calculated distances of the structure's coordinates regarding node 48 are between 4.5 Å and 4.6 Å (while the corresponding calculated distances from node 106 are larger than 4.6 Å), and can now be pruned since they are not found in the list of available distances. As for node 106, nothing can be done since their calculated distances can not be compared to what's available in the distance list.

In figures 77, 76, 75, 74, 73, 72, 71, 70, 69, 68, 67, 66, 65, 64 and 63 all solutions for the instance having the cutoff value of 4.5 Å are depicted. The solutions for the cutoff values of 4.6 Å and 4.9 Å are subsets of this set of solutions and can be identified by the remaining leafs in the trees from figures 60 and 61.

Each solution par is shown side by side as to illustrate their similarity. They are essentially identical except for their last part, highlighted in blue in each picture. This part is of the type of the second tetrahedron presented in figure 36, and the coordinates for their atoms are generated by the same distance.

These structures are the symmetric solutions resulted from the equation of the intersection of three spheres (equation 7.1) for the determination of the last hydrogen in the protein: the hydrogen coordinates are symmetric and, consequentially, the placement of each structure becomes symmetric to each other.

Interestingly, although these solution pairs are almost the same (having the differences just previously explained), one element of these pairs can be pruned when the cutoff value increases. For instance, in the pair 54 and 55 (figure 68) the solution 55 is pruned when the cutoff value goes from 4.6 Å to 4.9 Å. This happens due to the fact that a larger cutoff value implies more information (that is, more available distances for testing in the prune process).

In order to better analyze the behavior of the selected solutions according to the cutoff value, attached to each figure of the solution pairs (and cited in their caption) are the distances used in each structure, along with their respective hydrogen pairs. In black are the distances that were never analyzed in any of the three cutoff value's tests due to the fact that they were greater than 4.9 Å.

In light blue are the distances analyzed when the cutoff vale is equal to 4.5 Å, while in green are the additional distances used when the cutoff value is equal to 4.6 Å (and when the solutions are still present when the cutoff value is increased). This same rational is repeated for the distances in dark blue, which are used when the cutoff value

increases to 4.9 Å. In the pruning process, these distances can be calculated and compared to the list of available distances from the instance, and the candidate can be either pruned or not.

It can be seen that, although the distances are the same in the solutions, different solution pairs use these distances in different orders. Moreover, the different calculated distances from hydrogen i to hydrogen j, $j \leqslant i-3$, determine which solutions remain viable as the cutoff value increases and there is more information for the pruning process.

Although it can happen that all these additional calculated distances still remain above the cutoff value and can't be pruned, if by the end of the algorithm the situation arises where there is still some distance that remained without use, it it easy to conclude that this candidate cannot be the right solution and hence it is pruned at this stage.

In tables 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37 and 40, the distances are presented in the order they were used. In black are the distances used when the cutoff vale is equal to 4.5 Å, while in green are the additional distances used when the cutoff value is equal to 4.6 Å (and when the solutions are still present when the cutoff value is increased). The distances in blue are used when the cutoff value increases to 4.9 Å.

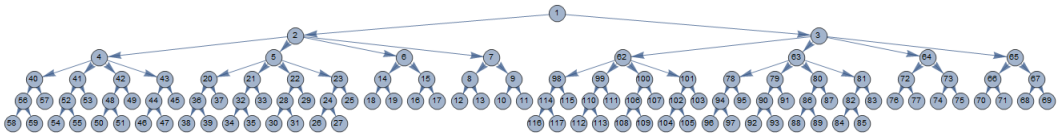

Figure 59 – Search tree for the instance with cutoff value equal to 4.5 Å.

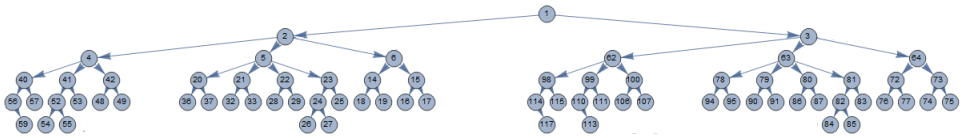

Figure 60 – Search tree for the instance with cutoff value equal to 4.6 Å.

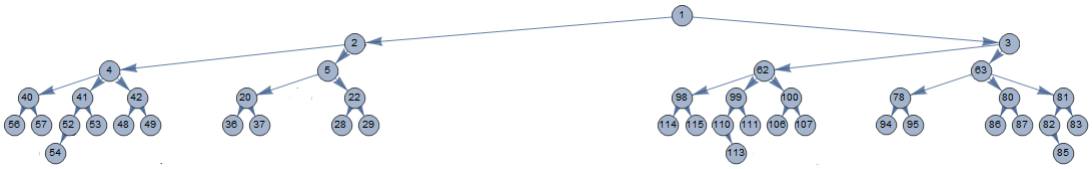

Figure 61 – Search tree for the instance with cutoff value equal to 4.9 Å.

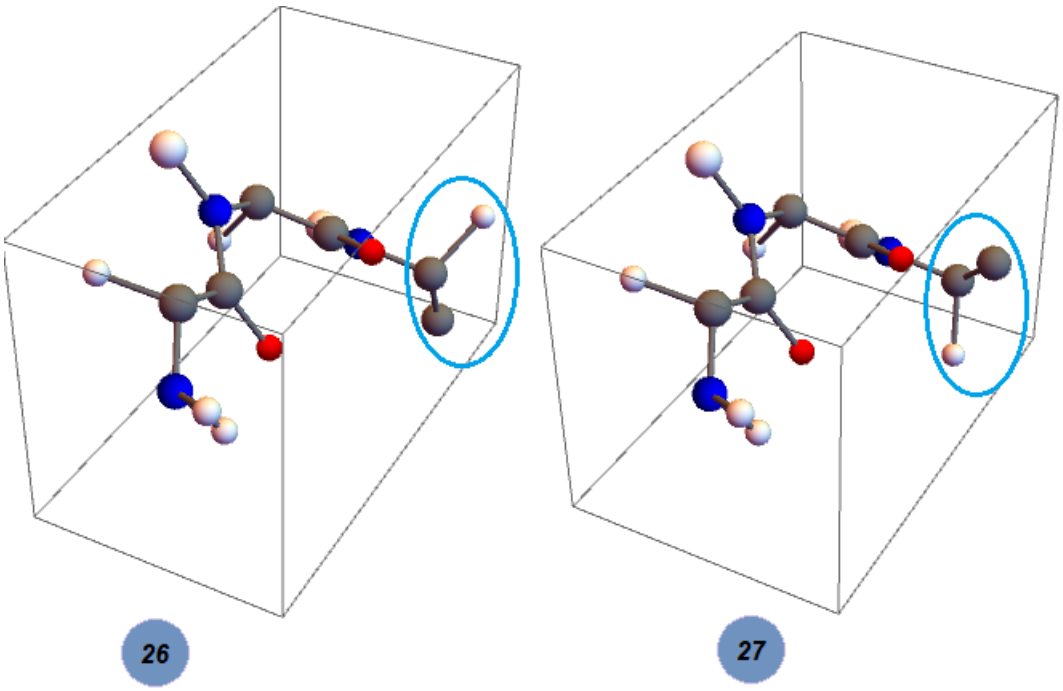

Figure 62 – Solutions for vertices 26 and 27 of the search tree. Distance list for vertex 26 is found in 7.4 and distance list for vertex 27 is found in 7.5

Vertex 26		**Vertex 27**	
Hydrogen Pair	Distance	Hydrogen Pair	Distance
(H_1,H_2)	2.49139	(H_1,H_2)	2.49139
(H_2,H_3)	3.52491	(H_2,H_3)	3.52491
(H_1,H_3)	*3.59369*	*(H_1,H_3)*	*3.59369*
(H_3,H_4)	2.65238	(H_3,H_4)	2.65238
(H_4,H_5)	3.6266	(H_4,H_5)	3.6266
(H_3,H_5)	*4.58485*	*(H_3,H_5)*	*4.58485*
(H_2,H_5)	*4.85149*	*(H_2,H_5)*	*4.85149*
(H_5,H_6)	2.53149	(H_5,H_6)	2.53149
(H_6,H_7)	3.43621	(H_6,H_7)	3.43621
-		*(H_5,H_7)*	*4.62605*

Table 25 – Comparison of distances between solutions 26 and 27 (shown in figure 62). The distances used in the prune are underlined. Distances used for the cutoff value of 4.6 Å are in green and additional distances used for the cutoff value of 4.9 Å are in blue.

$$
\begin{array}{llll}
(H_1, H_2) \rightarrow 2.49139 & (H_2, H_3) \rightarrow 3.52491 & (H_3, H_4) \rightarrow 2.65238 & (H_4, H_5) \rightarrow 3.6266 \\
(H_5, H_6) \rightarrow 2.53149 & (H_6, H_7) \rightarrow 3.43621 & (H_1, H_3) \rightarrow 3.59369 & (H_2, H_4) \rightarrow 5.0736 \\
(H_3, H_5) \rightarrow 4.58485 & (H_4, H_6) \rightarrow 5.58578 & (H_5, H_7) \rightarrow 5.20159 & (H_1, H_4) \rightarrow 5.28655 \\
(H_2, H_5) \rightarrow 4.85149 & (H_3, H_6) \rightarrow 6.91666 & (H_4, H_7) \rightarrow 6.88515 & (H_1, H_5) \rightarrow 6.53009 \\
(H_2, H_6) \rightarrow 6.45938 & (H_3, H_7) \rightarrow 8.34712 & (H_1, H_6) \rightarrow 8.18851 & (H_2, H_7) \rightarrow 7.32582 \\
(H_1, H_7) \rightarrow 8.43563 & & &
\end{array}
\tag{7.4}
$$

$$
\begin{array}{llll}
(H_1, H_2) \rightarrow 2.49139 & (H_2, H_3) \rightarrow 3.52491 & (H_3, H_4) \rightarrow 2.65238 & (H_4, H_5) \rightarrow 3.6266 \\
(H_5, H_6) \rightarrow 2.53149 & (H_6, H_7) \rightarrow 3.43621 & (H_1, H_3) \rightarrow 3.59369 & (H_2, H_4) \rightarrow 5.0736 \\
(H_3, H_5) \rightarrow 4.58485 & (H_4, H_6) \rightarrow 5.58578 & (H_5, H_7) \rightarrow 4.62605 & (H_1, H_4) \rightarrow 5.28655 \\
(H_2, H_5) \rightarrow 4.85149 & (H_3, H_6) \rightarrow 6.91666 & (H_4, H_7) \rightarrow 7.00581 & (H_1, H_5) \rightarrow 6.53009 \\
(H_2, H_6) \rightarrow 6.45938 & (H_3, H_7) \rightarrow 7.576 & (H_1, H_6) \rightarrow 8.18851 & (H_2, H_7) \rightarrow 5.5569 \\
(H_1, H_7) \rightarrow 7.22468 & & &
\end{array}
\tag{7.5}
$$

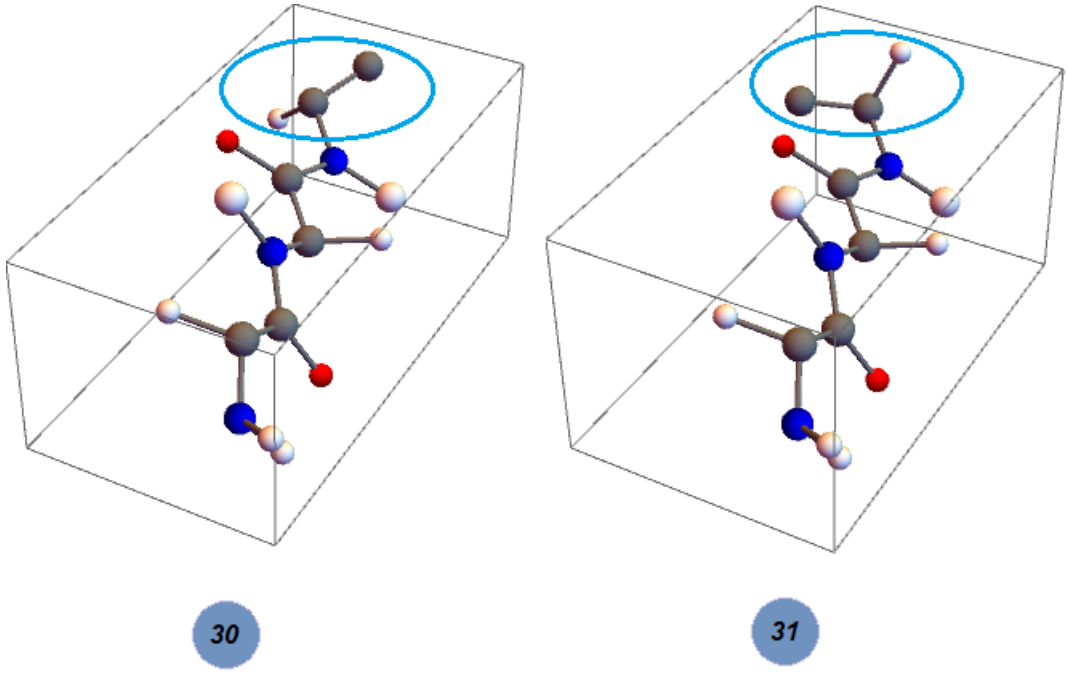

Figure 63 – Solutions for vertices 30 and 31 of the search tree. Distance list for vertex 30 is found in equation 7.6 and distance list for vertex 31 is found in equation 7.7

$$
\begin{array}{llll}
(H_1, H_2) \to 2.49139 & (H_2, H_3) \to 3.52491 & (H_3, H_4) \to 2.65238 & (H_4, H_5) \to 3.6266 \\
(H_5, H_6) \to 2.53149 & (H_6, H_7) \to 3.43621 & (H_1, H_3) \to 3.59369 & (H_2, H_4) \to 5.0736 \\
(H_3, H_5) \to 5.20287 & (H_4, H_6) \to 5.32866 & (H_5, H_7) \to 5.20159 & (H_1, H_4) \to 5.28655 \\
(H_2, H_5) \to 5.14169 & (H_3, H_6) \to 7.16445 & (H_4, H_7) \to 6.02986 & (H_1, H_5) \to 6.11035 \\
(H_2, H_6) \to 7.16653 & (H_3, H_7) \to 8.00764 & (H_1, H_6) \to 8.51733 & (H_2, H_7) \to 9.07461 \\
(H_1, H_7) \to 10.4737 &&&
\end{array}
\tag{7.6}
$$

$$
\begin{array}{llll}
(H_1, H_2) \to 2.49139 & (H_2, H_3) \to 3.52491 & (H_3, H_4) \to 2.65238 & (H_4, H_5) \to 3.6266 \\
(H_5, H_6) \to 2.53149 & (H_6, H_7) \to 3.43621 & (H_1, H_3) \to 3.59369 & (H_2, H_4) \to 5.0736 \\
(H_3, H_5) \to 5.20287 & (H_4, H_6) \to 5.32866 & (H_5, H_7) \to 4.62605 & (H_1, H_4) \to 5.28655 \\
(H_2, H_5) \to 5.14169 & (H_3, H_6) \to 7.16445 & (H_4, H_7) \to 5.51024 & (H_1, H_5) \to 6.11035 \\
(H_2, H_6) \to 7.16653 & (H_3, H_7) \to 8.02456 & (H_1, H_6) \to 8.51733 & (H_2, H_7) \to 9.346 \\
(H_1, H_7) \to 10.1544 &&&
\end{array}
\tag{7.7}
$$

Vertex 30		**Vertex 31**	
Hydrogen Pair	Distance	Hydrogen Pair	Distance
(H_1,H_2)	2.49139	(H_1,H_2)	2.49139
(H_2,H_3)	3.52491	(H_2,H_3)	3.52491
(H_1,H_3)	*3.59369*	*(H_1,H_3)*	*3.59369*
(H_3,H_4)	2.65238	(H_3,H_4)	2.65238
(H_4,H_5)	3.6266	(H_4,H_5)	3.6266
(H_5,H_6)	2.53149	(H_5,H_6)	2.53149
(H_6,H_7)	3.43621	(H_6,H_7)	3.43621
-		*(H_5,H_7)*	*4.62605*

Table 26 – Comparison of distances between solutions 30 and 31 (shown in figure 63). The distances used in the prune are underlined. Distances used for the cutoff value of 4.6 Å are in green and additional distances used for the cutoff value of 4.9 Å are in blue.

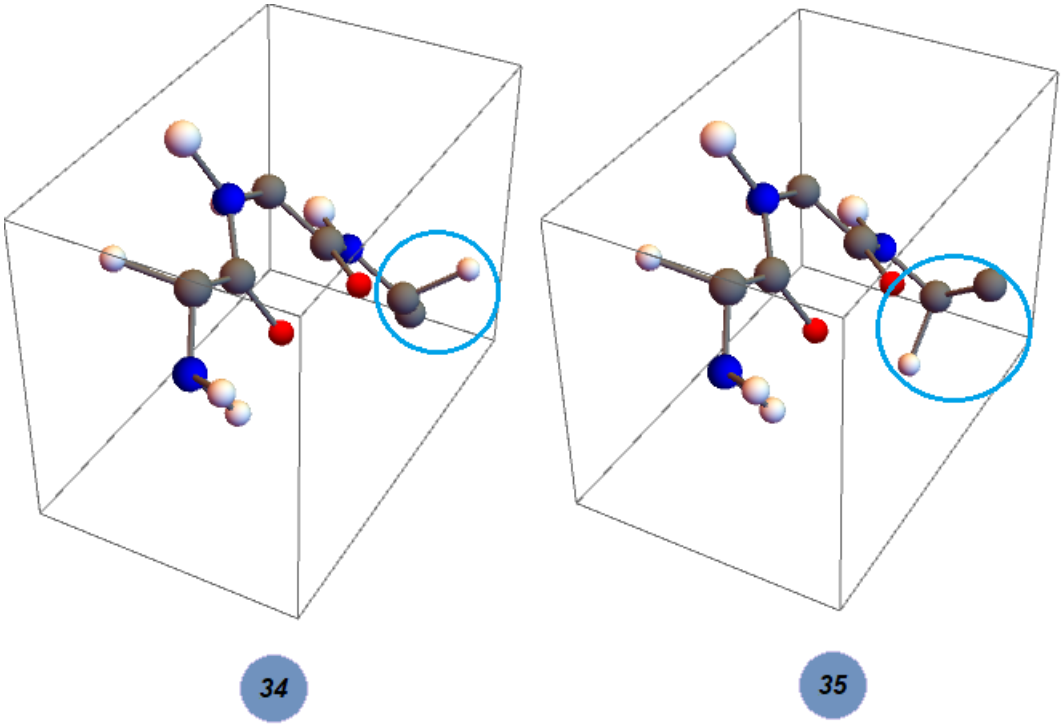

Figure 64 – Solutions for vertices 34 and 35 of the search tree. Distance list for vertex 34 is found in equation 7.8 and distance list for vertex 35 is found in equation 7.9.

Vertex 34		**Vertex 35**	
Hydrogen Pair	Distance	Hydrogen Pair	Distance
(H_1,H_2)	2.49139	(H_1,H_2)	2.49139
(H_2,H_3)	3.52491	(H_2,H_3)	3.52491
(H_1,H_3)	*3.59369*	*(H_1,H_3)*	*3.59369*
(H_3,H_4)	2.65238	(H_3,H_4)	2.65238
(H_4,H_5)	3.43621	(H_4,H_5)	3.43621
(H_3,H_5)	*4.63118*	*(H_3,H_5)*	*4.63118*
(H_5,H_6)	2.53149	(H_5,H_6)	2.53149
(H_6,H_7)	3.6266	(H_6,H_7)	3.6266
-		(H_5,H_7)	*4.59207*
-		*(H_2,H_7)*	*4.82612*

Table 27 – Comparison of distances between solutions 34 and 35 (shown in figure 64). The distances used in the prune are underlined. Distances used for the cutoff value of 4.6 Å are in green and additional distances used for the cutoff value of 4.9 Å are in blue.

$$
\begin{array}{llll}
(H_1, H_2) \rightarrow 2.49139 & (H_2, H_3) \rightarrow 3.52491 & (H_3, H_4) \rightarrow 2.65238 & (H_4, H_5) \rightarrow 3.43621 \\
(H_5, H_6) \rightarrow 2.53149 & (H_6, H_7) \rightarrow 3.6266 & (H_1, H_3) \rightarrow 3.59369 & (H_2, H_4) \rightarrow 5.0736 \\
(H_3, H_5) \rightarrow 4.63118 & (H_4, H_6) \rightarrow 5.54401 & (H_5, H_7) \rightarrow 5.15025 & (H_1, H_4) \rightarrow 5.28655 \\
(H_2, H_5) \rightarrow 5.24949 & (H_3, H_6) \rightarrow 6.89193 & (H_4, H_7) \rightarrow 6.79579 & (H_1, H_5) \rightarrow 6.80928 \\
(H_2, H_6) \rightarrow 6.54821 & (H_3, H_7) \rightarrow 7.84724 & (H_1, H_6) \rightarrow 8.29267 & (H_2, H_7) \rightarrow 6.34348 \\
(H_1, H_7) \rightarrow 7.65004 & & &
\end{array}
\tag{7.8}
$$

$$
\begin{array}{llll}
(H_1, H_2) \rightarrow 2.49139 & (H_2, H_3) \rightarrow 3.52491 & (H_3, H_4) \rightarrow 2.65238 & (H_4, H_5) \rightarrow 3.43621 \\
(H_5, H_6) \rightarrow 2.53149 & (H_6, H_7) \rightarrow 3.6266 & (H_1, H_3) \rightarrow 3.59369 & (H_2, H_4) \rightarrow 5.0736 \\
(H_3, H_5) \rightarrow 4.63118 & (H_4, H_6) \rightarrow 5.54401 & (H_5, H_7) \rightarrow 4.59207 & (H_1, H_4) \rightarrow 5.28655 \\
(H_2, H_5) \rightarrow 5.24949 & (H_3, H_6) \rightarrow 6.89193 & (H_4, H_7) \rightarrow 6.77707 & (H_1, H_5) \rightarrow 6.80928 \\
(H_2, H_6) \rightarrow 6.54821 & (H_3, H_7) \rightarrow 7.00579 & (H_1, H_6) \rightarrow 8.29267 & (H_2, H_7) \rightarrow 4.82612 \\
(H_1, H_7) \rightarrow 6.82534 & & &
\end{array}
\tag{7.9}
$$

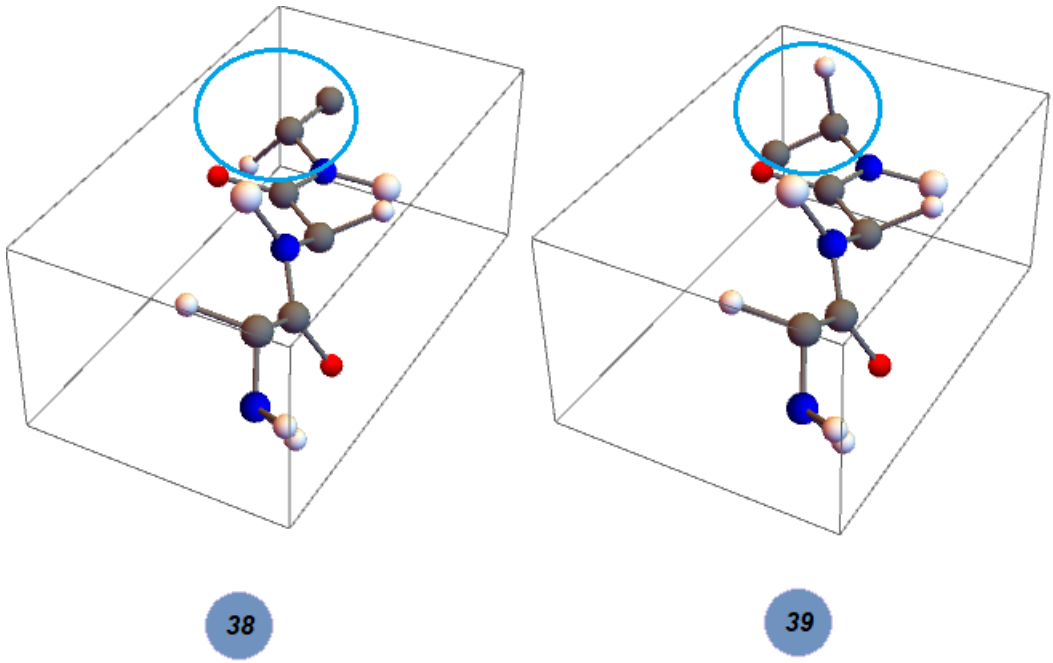

Figure 65 – Solutions for vertices 38 and 39 of the search tree. Distance list for vertex 38 is found in equation 7.10 and distance list for vertex 39 is found in equation 7.11.

$$
\begin{array}{llll}
(H_1, H_2) \to 2.49139 & (H_2, H_3) \to 3.52491 & (H_3, H_4) \to 2.65238 & (H_4, H_5) \to 3.43621 \\
(H_5, H_6) \to 2.53149 & (H_6, H_7) \to 3.6266 & (H_1, H_3) \to 3.59369 & (H_2, H_4) \to 5.0736 \\
(H_3, H_5) \to 5.26684 & (H_4, H_6) \to 5.27419 & (H_5, H_7) \to 5.15025 & (H_1, H_4) \to 5.28655 \\
(H_2, H_5) \to 5.52935 & (H_3, H_6) \to 7.15038 & (H_4, H_7) \to 5.84691 & (H_1, H_5) \to 6.39121 \\
(H_2, H_6) \to 7.27334 & (H_3, H_7) \to 7.43193 & (H_1, H_6) \to 8.6303 & (H_2, H_7) \to 8.4402 \\
(H_1, H_7) \to 10.0327 & & &
\end{array}
\tag{7.10}
$$

$$
\begin{array}{llll}
(H_1, H_2) \to 2.49139 & (H_2, H_3) \to 3.52491 & (H_3, H_4) \to 2.65238 & (H_4, H_5) \to 3.43621 \\
(H_5, H_6) \to 2.53149 & (H_6, H_7) \to 3.6266 & (H_1, H_3) \to 3.59369 & (H_2, H_4) \to 5.0736 \\
(H_3, H_5) \to 5.26684 & (H_4, H_6) \to 5.27419 & (H_5, H_7) \to 4.59207 & (H_1, H_4) \to 5.28655 \\
(H_2, H_5) \to 5.52935 & (H_3, H_6) \to 7.15038 & (H_4, H_7) \to 5.12442 & (H_1, H_5) \to 6.39121 \\
(H_2, H_6) \to 7.27334 & (H_3, H_7) \to 7.43193 & (H_1, H_6) \to 8.6303 & (H_2, H_7) \to 9.0596 \\
(H_1, H_7) \to 10.0546 & & &
\end{array}
\tag{7.11}
$$

Vertex 38		**Vertex 39**	
Hydrogen Pair	Distance	Hydrogen Pair	Distance
(H_1,H_2)	2.49139	(H_1,H_2)	2.49139
(H_2,H_3)	3.52491	(H_2,H_3)	3.52491
(H_1,H_3)	*3.59369*	(H_1,H_3)	*3.59369*
(H_3,H_4)	2.65238	(H_3,H_4)	2.65238
(H_4,H_5)	3.43621	(H_4,H_5)	3.43621
-		(H_3,H_5)	*4.63118*
(H_5,H_6)	2.53149	(H_5,H_6)	2.53149
(H_6,H_7)	3.6266	(H_6,H_7)	3.6266
-		(H_5,H_7)	*4.59207*

Table 28 – Comparison of distances between solutions 38 and 39 (shown in figure 65). The distances used in the prune are underlined. Distances used for the cutoff value of 4.6 Å are in green and additional distances used for the cutoff value of 4.9 Å are in blue.

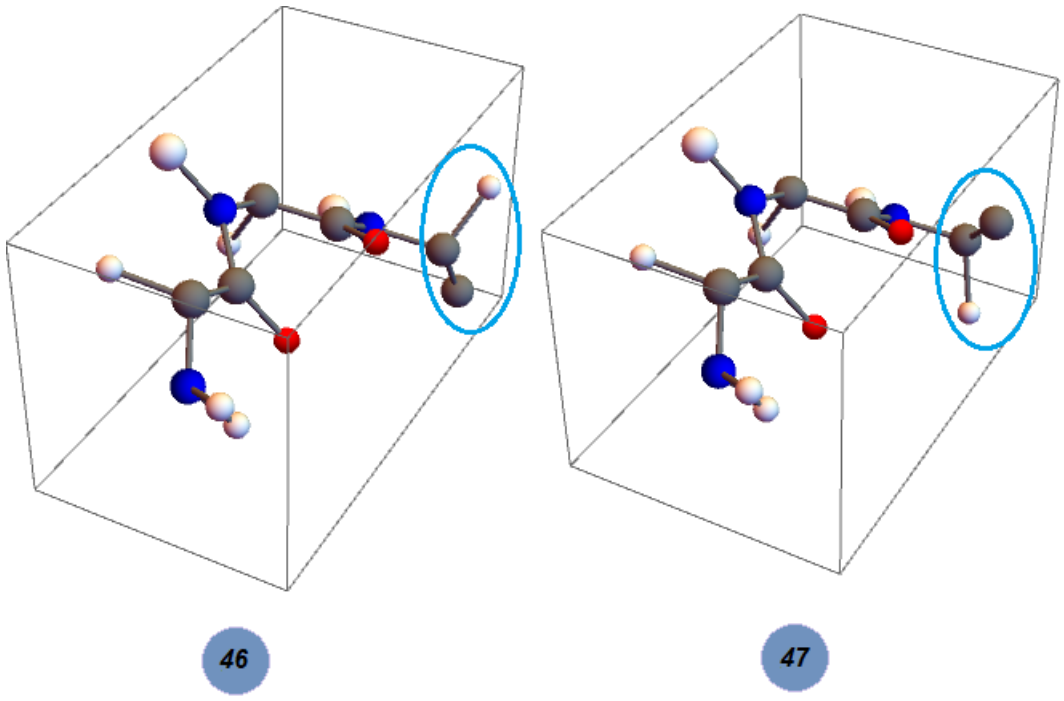

Figure 66 – Solutions for vertices 46 and 47 of the search tree. Distance list for vertex 46 is found in equation 7.12 and distance list for vertex 47 is found in equation 7.13.

Vertex 46		**Vertex 47**	
Hydrogen Pair	Distance	Hydrogen Pair	Distance
(H_1,H_2)	2.49139	(H_1,H_2)	2.49139
(H_2,H_3)	3.52491	(H_2,H_3)	3.52491
(H_1,H_3)	*3.59369*	*(H_1,H_3)*	*3.59369*
(H_3,H_4)	2.53149	(H_3,H_4)	2.53149
(H_4,H_5)	3.6266	(H_4,H_5)	3.6266
(H_3,H_5)	*4.59207*	*(H_3,H_5)*	*4.59207*
(H_2,H_5)	*4.76188*	*(H_2,H_5)*	*4.76188*
(H_5,H_6)	2.65238	(H_5,H_6)	2.65238
(H_6,H_7)	3.43621	(H_6,H_7)	3.43621
-		*(H_5,H_7)*	*4.63118*

Table 29 – Comparison of distances between solutions 46 and 47 (shown in figure 66). The distances used in the prune are underlined. Distances used for the cutoff value of 4.6 Å are in green and additional distances used for the cutoff value of 4.9 Å are in blue.

$$
\begin{array}{llll}
(H_1, H_2) \rightarrow 2.49139 & (H_2, H_3) \rightarrow 3.52491 & (H_3, H_4) \rightarrow 2.53149 & (H_4, H_5) \rightarrow 3.6266 \\
(H_5, H_6) \rightarrow 2.65238 & (H_6, H_7) \rightarrow 3.43621 & (H_1, H_3) \rightarrow 3.59369 & (H_2, H_4) \rightarrow 5.03461 \\
(H_3, H_5) \rightarrow 4.59207 & (H_4, H_6) \rightarrow 5.56355 & (H_5, H_7) \rightarrow 5.26684 & (H_1, H_4) \rightarrow 5.33484 \\
(H_2, H_5) \rightarrow 4.76188 & (H_3, H_6) \rightarrow 6.98832 & (H_4, H_7) \rightarrow 6.82448 & (H_1, H_5) \rightarrow 6.49867 \\
(H_2, H_6) \rightarrow 6.61767 & (H_3, H_7) \rightarrow 8.28216 & (H_1, H_6) \rightarrow 8.30505 & (H_2, H_7) \rightarrow 7.46832 \\
(H_1, H_7) \rightarrow 8.4855 & & &
\end{array}
\tag{7.12}
$$

$$
\begin{array}{llll}
(H_1, H_2) \rightarrow 2.49139 & (H_2, H_3) \rightarrow 3.52491 & (H_3, H_4) \rightarrow 2.53149 & (H_4, H_5) \rightarrow 3.6266 \\
(H_5, H_6) \rightarrow 2.65238 & (H_6, H_7) \rightarrow 3.43621 & (H_1, H_3) \rightarrow 3.59369 & (H_2, H_4) \rightarrow 5.03461 \\
(H_3, H_5) \rightarrow 4.59207 & (H_4, H_6) \rightarrow 5.56355 & (H_5, H_7) \rightarrow 4.63118 & (H_1, H_4) \rightarrow 5.33484 \\
(H_2, H_5) \rightarrow 4.76188 & (H_3, H_6) \rightarrow 6.98832 & (H_4, H_7) \rightarrow 7.04444 & (H_1, H_5) \rightarrow 6.49867 \\
(H_2, H_6) \rightarrow 6.61767 & (H_3, H_7) \rightarrow 7.70401 & (H_1, H_6) \rightarrow 8.30505 & (H_2, H_7) \rightarrow 5.83569 \\
(H_1, H_7) \rightarrow 7.41799 & & &
\end{array}
\tag{7.13}
$$

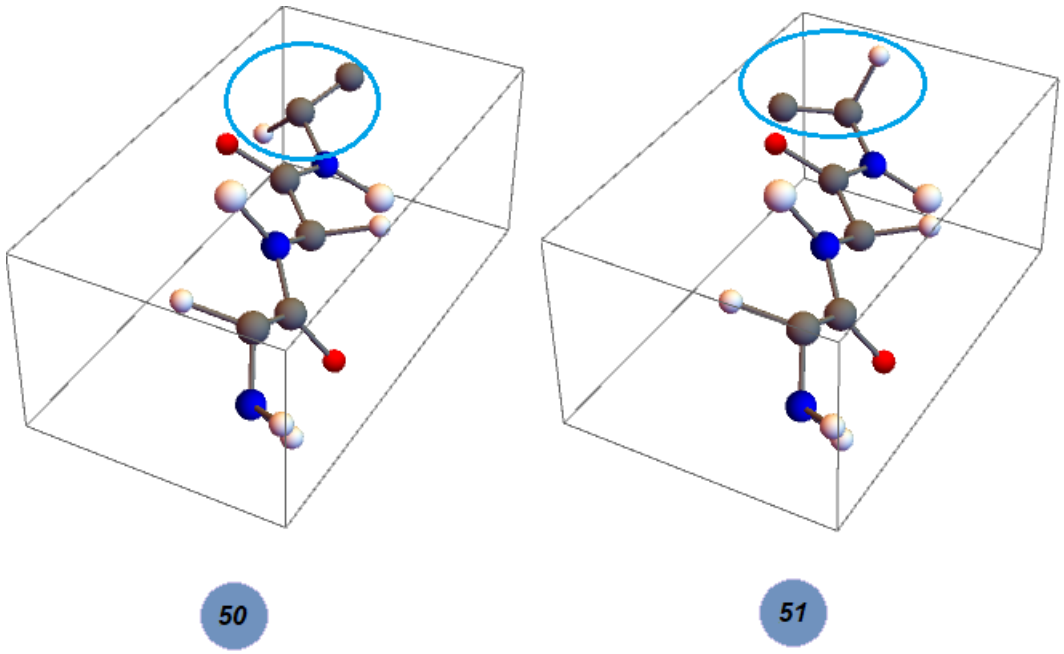

Figure 67 – Solutions for vertices 50 and 51 of the search tree. Distance list for vertex 50 is found in equation 7.14 and distance list for vertex 51 is found in equation 7.15.

$$
\begin{array}{llll}
(H_1, H_2) \to 2.49139 & (H_2, H_3) \to 3.52491 & (H_3, H_4) \to 2.53149 & (H_4, H_5) \to 3.6266 \\
(H_5, H_6) \to 2.65238 & (H_6, H_7) \to 3.43621 & (H_1, H_3) \to 3.59369 & (H_2, H_4) \to 5.03461 \\
(H_3, H_5) \to 5.15025 & (H_4, H_6) \to 5.37946 & (H_5, H_7) \to 5.26684 & (H_1, H_4) \to 5.33484 \\
(H_2, H_5) \to 5.19592 & (H_3, H_6) \to 7.11798 & (H_4, H_7) \to 5.98868 & (H_1, H_5) \to 6.17555 \\
(H_2, H_6) \to 7.12289 & (H_3, H_7) \to 7.88446 & (H_1, H_6) \to 8.58179 & (H_2, H_7) \to 8.93543 \\
(H_1, H_7) \to 10.4408 &&&
\end{array}
\tag{7.14}
$$

$$
\begin{array}{llll}
(H_1, H_2) \to 2.49139 & (H_2, H_3) \to 3.52491 & (H_3, H_4) \to 2.53149 & (H_4, H_5) \to 3.6266 \\
(H_5, H_6) \to 2.65238 & (H_6, H_7) \to 3.43621 & (H_1, H_3) \to 3.59369 & (H_2, H_4) \to 5.03461 \\
(H_3, H_5) \to 5.15025 & (H_4, H_6) \to 5.37946 & (H_5, H_7) \to 4.63118 & (H_1, H_4) \to 5.33484 \\
(H_2, H_5) \to 5.19592 & (H_3, H_6) \to 7.11798 & (H_4, H_7) \to 5.61175 & (H_1, H_5) \to 6.17555 \\
(H_2, H_6) \to 7.12289 & (H_3, H_7) \to 8.01035 & (H_1, H_6) \to 8.58179 & (H_2, H_7) \to 9.34248 \\
(H_1, H_7) \to 10.289 &&&
\end{array}
\tag{7.15}
$$

Vertex 50		**Vertex 51**	
Hydrogen Pair	Distance	Hydrogen Pair	Distance
(H_1,H_2)	2.49139	(H_1,H_2)	2.49139
(H_2,H_3)	3.52491	(H_2,H_3)	3.52491
(H_1,H_3)	*3.59369*	*(H_1,H_3)*	*3.59369*
(H_3,H_4)	2.53149	(H_3,H_4)	2.53149
(H_4,H_5)	3.6266	(H_4,H_5)	3.6266
(H_5,H_6)	2.65238	(H_5,H_6)	2.65238
(H_6,H_7)	3.43621	(H_6,H_7)	3.43621
-		*(H_5,H_7)*	*4.63118*

Table 30 – Comparison of distances between solutions 50 and 51 (shown in figure 67). The distances used in the prune are underlined. Distances used for the cutoff value of 4.6 Å are in green and additional distances used for the cutoff value of 4.9 Å are in blue.

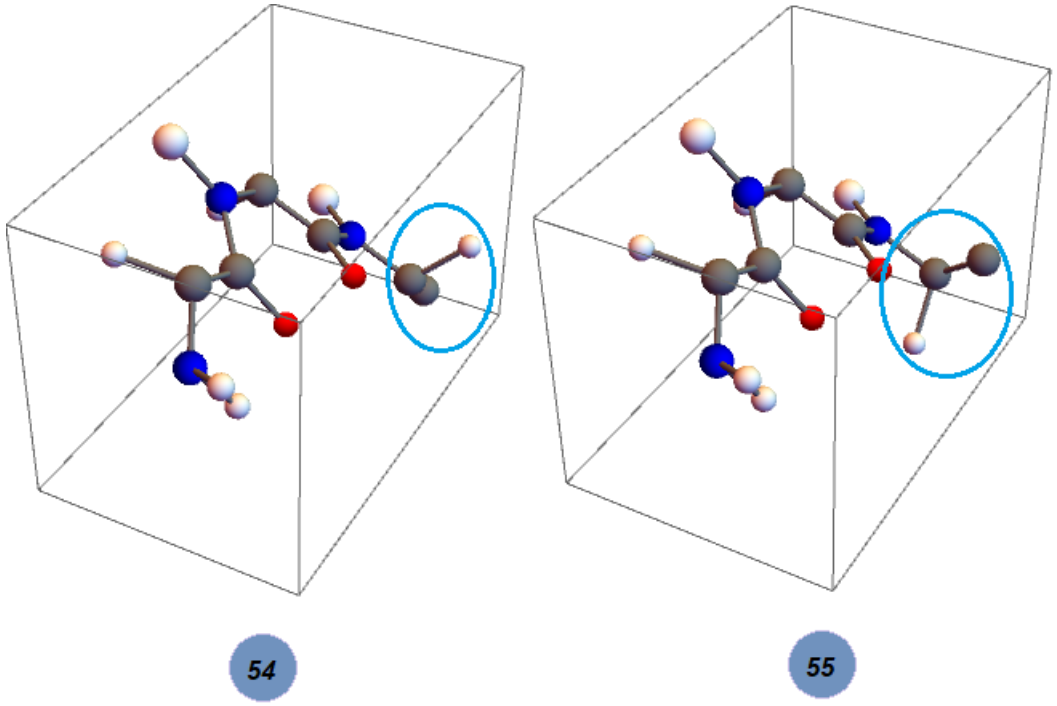

Figure 68 – Solutions for vertices 54 and 55 of the search tree. Distance list for vertex 54 is found in equation 7.16 and distance list for vertex 55 is found in equation 7.17.

Vertex 54		**Vertex 55**	
Hydrogen Pair	Distance	Hydrogen Pair	Distance
(H_1,H_2)	2.49139	(H_1,H_2)	2.49139
(H_2,H_3)	3.52491	(H_2,H_3)	3.52491
(H_1,H_3)	*3.59369*	*(H_1,H_3)*	*3.59369*
(H_3,H_4)	2.53149	(H_3,H_4)	2.53149
(H_4,H_5)	3.43621	(H_4,H_5)	3.43621
(H_3,H_5)	*4.62605*	*(H_3,H_5)*	*4.62605*
(H_5,H_6)	2.65238	(H_5,H_6)	2.65238
(H_6,H_7)	3.6266	(H_6,H_7)	3.6266
(H_5,H_7)	*4.58485*	*(H_5,H_7)*	*4.58485*

Table 31 – Comparison of distances between solutions 54 and 55 (shown in figure 68). The distances used in the prune are underlined. Distances used for the cutoff value of 4.6 Å are in green and additional distances used for the cutoff value of 4.9 Å are in blue.

$$
\begin{array}{llll}
(H_1, H_2) \rightarrow 2.49139 & (H_2, H_3) \rightarrow 3.52491 & (H_3, H_4) \rightarrow 2.53149 & (H_4, H_5) \rightarrow 3.43621 \\
(H_5, H_6) \rightarrow 2.65238 & (H_6, H_7) \rightarrow 3.6266 & (H_1, H_3) \rightarrow 3.59369 & (H_2, H_4) \rightarrow 5.03461 \\
(H_3, H_5) \rightarrow 4.62605 & (H_4, H_6) \rightarrow 5.54055 & (H_5, H_7) \rightarrow 5.20287 & (H_1, H_4) \rightarrow 5.33484 \\
(H_2, H_5) \rightarrow 5.15635 & (H_3, H_6) \rightarrow 6.96997 & (H_4, H_7) \rightarrow 6.74999 & (H_1, H_5) \rightarrow 6.78496 \\
(H_2, H_6) \rightarrow 6.65475 & (H_3, H_7) \rightarrow 7.79182 & (H_1, H_6) \rightarrow 8.36265 & (H_2, H_7) \rightarrow 6.43289 \\
(H_1, H_7) \rightarrow 7.62596 & & &
\end{array}
\tag{7.16}
$$

$$
\begin{array}{llll}
(H_1, H_2) \rightarrow 2.49139 & (H_2, H_3) \rightarrow 3.52491 & (H_3, H_4) \rightarrow 2.53149 & (H_4, H_5) \rightarrow 3.43621 \\
(H_5, H_6) \rightarrow 2.65238 & (H_6, H_7) \rightarrow 3.6266 & (H_1, H_3) \rightarrow 3.59369 & (H_2, H_4) \rightarrow 5.03461 \\
(H_3, H_5) \rightarrow 4.62605 & (H_4, H_6) \rightarrow 5.54055 & (H_5, H_7) \rightarrow 4.58485 & (H_1, H_4) \rightarrow 5.33484 \\
(H_2, H_5) \rightarrow 5.15635 & (H_3, H_6) \rightarrow 6.96997 & (H_4, H_7) \rightarrow 6.82054 & (H_1, H_5) \rightarrow 6.78496 \\
(H_2, H_6) \rightarrow 6.65475 & (H_3, H_7) \rightarrow 7.15323 & (H_1, H_6) \rightarrow 8.36265 & (H_2, H_7) \rightarrow 5.03576 \\
(H_1, H_7) \rightarrow 6.94711 & & &
\end{array}
\tag{7.17}
$$

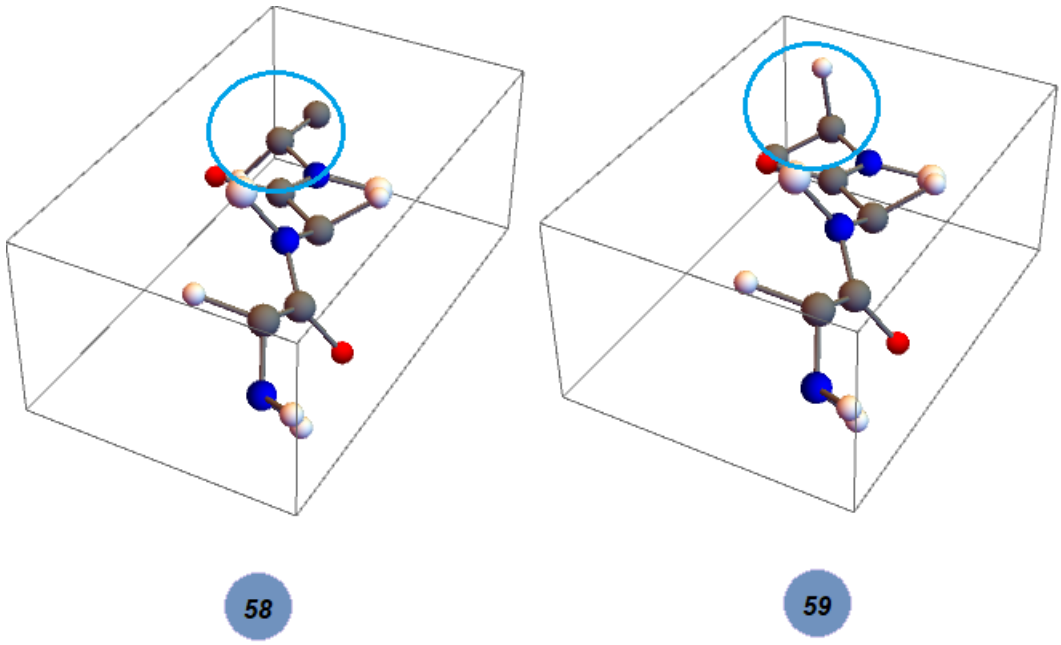

Figure 69 – Solutions for vertices 58 and 59 of the search tree. Distance list for vertex 58 is found in equation 7.18 and distance list for vertex 59 is found in equation 7.19.

$$
\begin{array}{llll}
(H_1, H_2) \to 2.49139 & (H_2, H_3) \to 3.52491 & (H_3, H_4) \to 2.53149 & (H_4, H_5) \to 3.43621 \\
(H_5, H_6) \to 2.65238 & (H_6, H_7) \to 3.6266 & (H_1, H_3) \to 3.59369 & (H_2, H_4) \to 5.03461 \\
(H_3, H_5) \to 5.20159 & (H_4, H_6) \to 5.34811 & (H_5, H_7) \to 5.20287 & (H_1, H_4) \to 5.33484 \\
(H_2, H_5) \to 5.57526 & (H_3, H_6) \to 7.10513 & (H_4, H_7) \to 5.80997 & (H_1, H_5) \to 6.46344 \\
(H_2, H_6) \to 7.1768 & (H_3, H_7) \to 7.33043 & (H_1, H_6) \to 8.64839 & (H_2, H_7) \to 8.24638 \\
(H_1, H_7) \to 9.94343 & & &
\end{array}
\tag{7.18}
$$

$$
\begin{array}{llll}
(H_1, H_2) \to 2.49139 & (H_2, H_3) \to 3.52491 & (H_3, H_4) \to 2.53149 & (H_4, H_5) \to 3.43621 \\
(H_5, H_6) \to 2.65238 & (H_6, H_7) \to 3.6266 & (H_1, H_3) \to 3.59369 & (H_2, H_4) \to 5.03461 \\
(H_3, H_5) \to 5.20159 & (H_4, H_6) \to 5.34811 & (H_5, H_7) \to 4.58485 & (H_1, H_4) \to 5.33484 \\
(H_2, H_5) \to 5.57526 & (H_3, H_6) \to 7.10513 & (H_4, H_7) \to 5.22397 & (H_1, H_5) \to 6.46344 \\
(H_2, H_6) \to 7.1768 & (H_3, H_7) \to 7.44546 & (H_1, H_6) \to 8.64839 & (H_2, H_7) \to 9.00778 \\
(H_1, H_7) \to 10.1394 & & &
\end{array}
\tag{7.19}
$$

Vertex 58		**Vertex 59**	
Hydrogen Pair	Distance	Hydrogen Pair	Distance
(H_1,H_2)	2.49139	(H_1,H_2)	2.49139
(H_2,H_3)	3.52491	(H_2,H_3)	3.52491
(H_1,H_3)	*3.59369*	*(H_1,H_3)*	*3.59369*
(H_3,H_4)	2.53149	(H_3,H_4)	2.53149
(H_4,H_5)	3.43621	(H_4,H_5)	3.43621
(H_5,H_6)	2.65238	(H_5,H_6)	2.65238
(H_6,H_7)	3.6266	(H_6,H_7)	3.6266
-		*(H_5,H_7)*	*4.58485*

Table 32 – Comparison of distances between solutions 58 and 59 (shown in figure 69). The distances used in the prune are underlined. Distances used for the cutoff value of 4.6 Å are in green and additional distances used for the cutoff value of 4.9 Å are in blue.

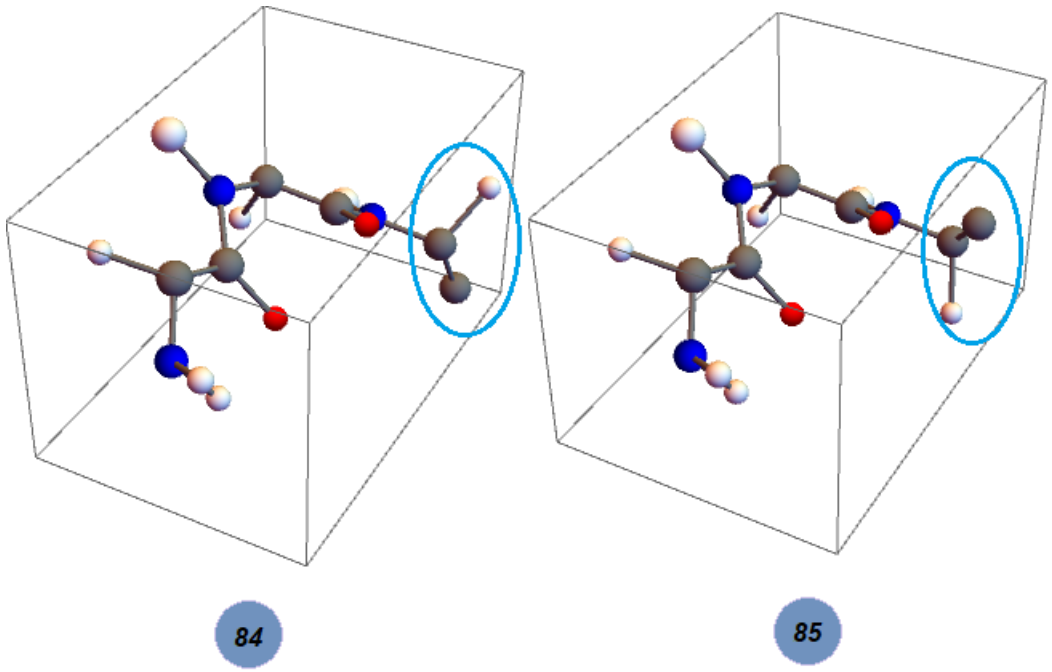

Figure 70 – Solutions for vertices 84 and 85 of the search tree. Distance list for vertex 84 is found in equation 7.20 and distance list for vertex 85 is found in equation 7.21.

Vertex 84		**Vertex 85**	
Hydrogen Pair	Distance	Hydrogen Pair	Distance
(H_1,H_2)	2.49139	(H_1,H_2)	2.49139
(H_2,H_3)	3.52491	(H_2,H_3)	3.52491
(H_1,H_3)	*3.59369*	*(H_1,H_3)*	*3.59369*
(H_3,H_4)	2.65238	(H_3,H_4)	2.65238
(H_4,H_5)	3.6266	(H_4,H_5)	3.6266
(H_3,H_5)	*4.58485*	*(H_3,H_5)*	*4.58485*
(H_5,H_6)	2.53149	(H_5,H_6)	2.53149
(H_6,H_7)	3.43621	(H_6,H_7)	3.43621
-		(H_5,H_7)	*4.62605*

Table 33 – Comparison of distances between solutions 84 and 85 (shown in figure 70). The distances used in the prune are underlined. Distances used for the cutoff value of 4.6 Å are in green and additional distances used for the cutoff value of 4.9 Å are in blue.

$$
\begin{array}{llll}
(H_1,H_2) \to 2.49139 & (H_2,H_3) \to 3.52491 & (H_3,H_4) \to 2.65238 & (H_4,H_5) \to 3.6266 \\
(H_5,H_6) \to 2.53149 & (H_6,H_7) \to 3.43621 & (H_1,H_3) \to 3.59369 & (H_2,H_4) \to 5.0736 \\
(H_3,H_5) \to 4.58485 & (H_4,H_6) \to 5.58578 & (H_5,H_7) \to 5.20159 & (H_1,H_4) \to 5.28655 \\
(H_2,H_5) \to 4.85149 & (H_3,H_6) \to 6.91666 & (H_4,H_7) \to 6.88515 & (H_1,H_5) \to 6.53009 \\
(H_2,H_6) \to 6.45938 & (H_3,H_7) \to 8.34712 & (H_1,H_6) \to 8.18851 & (H_2,H_7) \to 7.32582 \\
(H_1,H_7) \to 8.43563 & & &
\end{array}
\tag{7.20}
$$

$$
\begin{array}{llll}
(H_1,H_2) \to 2.49139 & (H_2,H_3) \to 3.52491 & (H_3,H_4) \to 2.65238 & (H_4,H_5) \to 3.6266 \\
(H_5,H_6) \to 2.53149 & (H_6,H_7) \to 3.43621 & (H_1,H_3) \to 3.59369 & (H_2,H_4) \to 5.0736 \\
(H_3,H_5) \to 4.58485 & (H_4,H_6) \to 5.58578 & (H_5,H_7) \to 4.62605 & (H_1,H_4) \to 5.28655 \\
(H_2,H_5) \to 4.85149 & (H_3,H_6) \to 6.91666 & (H_4,H_7) \to 7.0058 & (H_1,H_5) \to 6.53009 \\
(H_2,H_6) \to 6.45938 & (H_3,H_7) \to 7.576 & (H_1,H_6) \to 8.18851 & (H_2,H_7) \to 5.5569 \\
(H_1,H_7) \to 7.22468 & & &
\end{array}
\tag{7.21}
$$

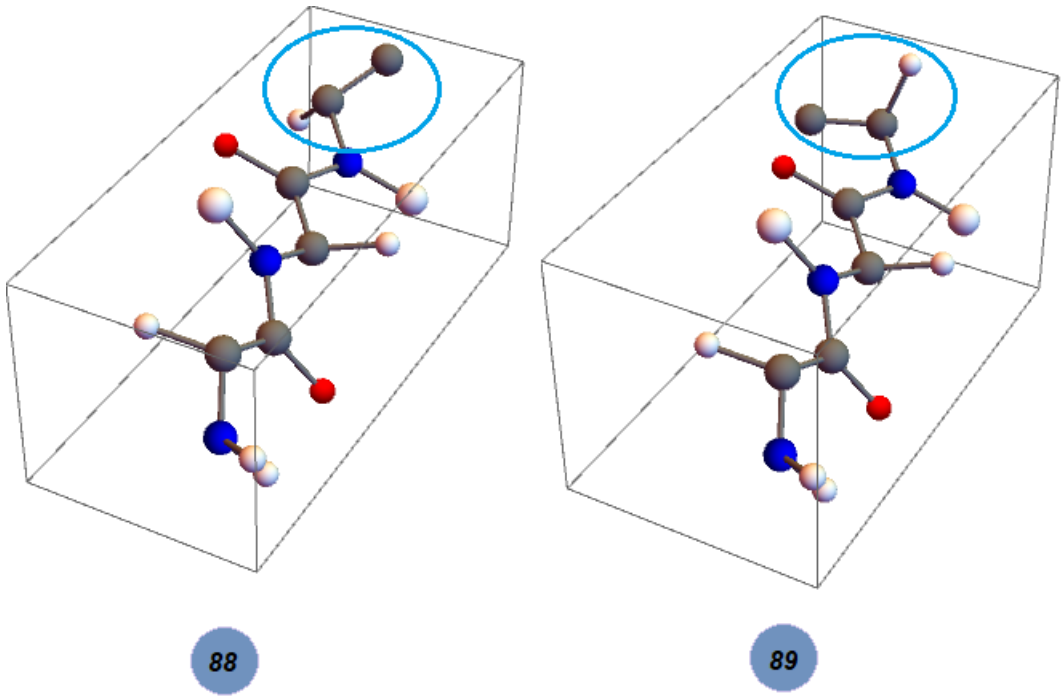

Figure 71 – Solutions for vertices 88 and 89 of the search tree. Distance list for vertex 88 is found in equation 7.22 and distance list for vertex 89 is found in equation 7.23.

$$
\begin{array}{llll}
(H_1, H_2) \to 2.49139 & (H_2, H_3) \to 3.52491 & (H_3, H_4) \to 2.65238 & (H_4, H_5) \to 3.6266 \\
(H_5, H_6) \to 2.53149 & (H_6, H_7) \to 3.43621 & (H_1, H_3) \to 3.59369 & (H_2, H_4) \to 5.0736 \\
(H_3, H_5) \to 5.20287 & (H_4, H_6) \to 5.32866 & (H_5, H_7) \to 5.20159 & (H_1, H_4) \to 5.28655 \\
(H_2, H_5) \to 5.14169 & (H_3, H_6) \to 7.16445 & (H_4, H_7) \to 6.02986 & (H_1, H_5) \to 6.11035 \\
(H_2, H_6) \to 7.16653 & (H_3, H_7) \to 8.00764 & (H_1, H_6) \to 8.51733 & (H_2, H_7) \to 9.07461 \\
(H_1, H_7) \to 10.4737 &&&
\end{array}
\tag{7.22}
$$

$$
\begin{array}{llll}
(H_1, H_2) \to 2.49139 & (H_2, H_3) \to 3.52491 & (H_3, H_4) \to 2.65238 & (H_4, H_5) \to 3.6266 \\
(H_5, H_6) \to 2.53149 & (H_6, H_7) \to 3.43621 & (H_1, H_3) \to 3.59369 & (H_2, H_4) \to 5.0736 \\
(H_3, H_5) \to 5.20287 & (H_4, H_6) \to 5.32866 & (H_5, H_7) \to 4.62605 & (H_1, H_4) \to 5.28655 \\
(H_2, H_5) \to 5.14169 & (H_3, H_6) \to 7.16445 & (H_4, H_7) \to 5.51024 & (H_1, H_5) \to 6.11035 \\
(H_2, H_6) \to 7.16653 & (H_3, H_7) \to 8.02456 & (H_1, H_6) \to 8.51733 & (H_2, H_7) \to 9.346 \\
(H_1, H_7) \to 10.1544 &&&
\end{array}
\tag{7.23}
$$

Vertex 88		**Vertex 89**	
Hydrogen Pair	Distance	Hydrogen Pair	Distance
(H_1,H_2)	2.49139	(H_1,H_2)	2.49139
(H_2,H_3)	3.52491	(H_2,H_3)	3.52491
(H_1,H_3)	*3.59369*	*(H_1,H_3)*	*3.59369*
(H_3,H_4)	2.65238	(H_3,H_4)	2.65238
(H_4,H_5)	3.6266	(H_4,H_5)	3.6266
(H_5,H_6)	2.53149	(H_5,H_6)	2.53149
(H_6,H_7)	3.43621	(H_6,H_7)	3.43621
-		(H_5,H_7)	*4.62605*

Table 34 – Comparison of distances between solutions 88 and 89 (shown in figure 71). The distances used in the prune are underlined. Distances used for the cutoff value of 4.6 Å are in green and additional distances used for the cutoff value of 4.9 Å are in blue.

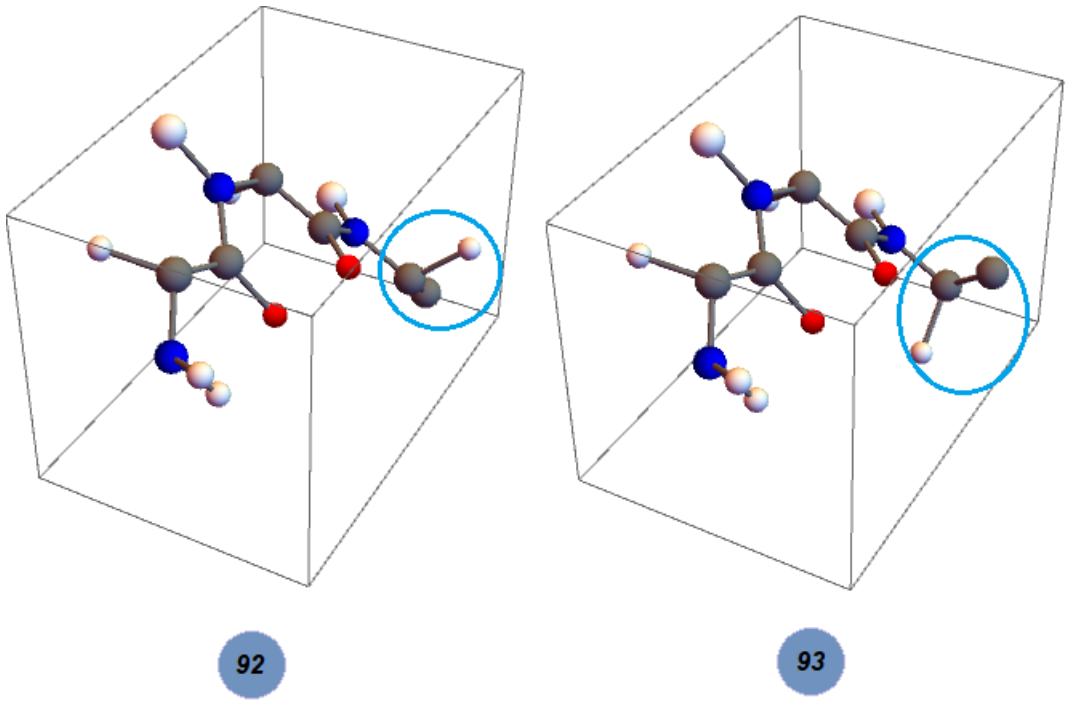

Figure 72 – Solutions for vertices 92 and 93 of the search tree. Distance list for vertex 92 is found in equation 7.24 and distance list for vertex 93 is found in equation 7.25.

Vertex 92		**Vertex 93**	
Hydrogen Pair	Distance	Hydrogen Pair	Distance
(H_1,H_2)	2.49139	(H_1,H_2)	2.49139
(H_2,H_3)	3.52491	(H_2,H_3)	3.52491
(H_1,H_3)	*3.59369*	*(H_1,H_3)*	*3.59369*
(H_3,H_4)	2.65238	(H_3,H_4)	2.65238
(H_4,H_5)	3.43621	(H_4,H_5)	3.43621
(H_3,H_5)	*4.63118*	*(H_3,H_5)*	*4.63118*
(H_5,H_6)	2.53149	(H_5,H_6)	2.53149
(H_6,H_7)	3.6266	(H_6,H_7)	3.6266
-		*(H_2,H_7)*	*4.82612*
-		*(H_5,H_7)*	*4.59207*

Table 35 – Comparison of distances between solutions 92 and 93 (shown in figure 72). The distances used in the prune are underlined. Distances used for the cutoff value of 4.6 Å are in green and additional distances used for the cutoff value of 4.9 Å are in blue.

$$
\begin{array}{llll}
(H_1, H_2) \rightarrow 2.49139 & (H_2, H_3) \rightarrow 3.52491 & (H_3, H_4) \rightarrow 2.65238 & (H_4, H_5) \rightarrow 3.43621 \\
(H_5, H_6) \rightarrow 2.53149 & (H_6, H_7) \rightarrow 3.6266 & (H_1, H_3) \rightarrow 3.59369 & (H_2, H_4) \rightarrow 5.0736 \\
(H_3, H_5) \rightarrow 4.63118 & (H_4, H_6) \rightarrow 5.54401 & (H_5, H_7) \rightarrow 5.15025 & (H_1, H_4) \rightarrow 5.28655 \\
(H_2, H_5) \rightarrow 5.24949 & (H_3, H_6) \rightarrow 6.89193 & (H_4, H_7) \rightarrow 6.79579 & (H_1, H_5) \rightarrow 6.80928 \\
(H_2, H_6) \rightarrow 6.54821 & (H_3, H_7) \rightarrow 7.84724 & (H_1, H_6) \rightarrow 8.29267 & (H_2, H_7) \rightarrow 6.34348 \\
(H_1, H_7) \rightarrow 7.65004 & & &
\end{array}
\tag{7.24}
$$

$$
\begin{array}{llll}
(H_1, H_2) \rightarrow 2.49139 & (H_2, H_3) \rightarrow 3.52491 & (H_3, H_4) \rightarrow 2.65238 & (H_4, H_5) \rightarrow 3.43621 \\
(H_5, H_6) \rightarrow 2.53149 & (H_6, H_7) \rightarrow 3.6266 & (H_1, H_3) \rightarrow 3.59369 & (H_2, H_4) \rightarrow 5.0736 \\
(H_3, H_5) \rightarrow 4.63118 & (H_4, H_6) \rightarrow 5.54401 & (H_5, H_7) \rightarrow 4.59207 & (H_1, H_4) \rightarrow 5.28655 \\
(H_2, H_5) \rightarrow 5.24949 & (H_3, H_6) \rightarrow 6.89193 & (H_4, H_7) \rightarrow 6.77707 & (H_1, H_5) \rightarrow 6.80928 \\
(H_2, H_6) \rightarrow 6.54821 & (H_3, H_7) \rightarrow 7.00579 & (H_1, H_6) \rightarrow 8.29267 & (H_2, H_7) \rightarrow 4.82612 \\
(H_1, H_7) \rightarrow 6.82534 & & &
\end{array}
\tag{7.25}
$$

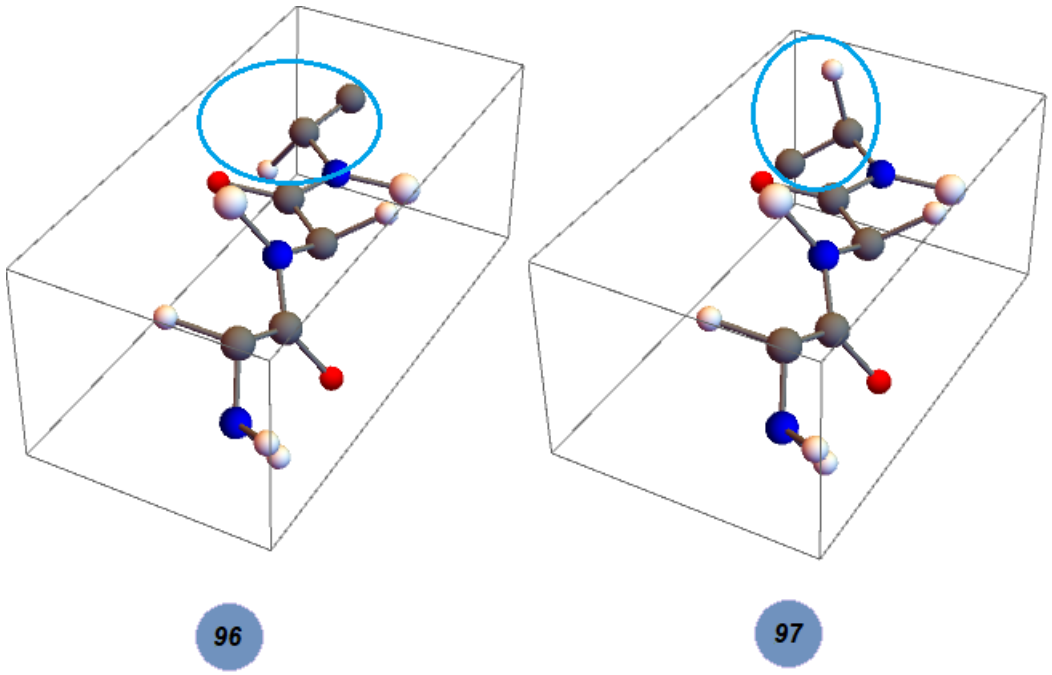

Figure 73 – Solutions for vertices 96 and 97 of the search tree. Distance list for vertex 96 is found in equation 7.26 and distance list for vertex 97 is found in equation 7.27.

$$
\begin{array}{llll}
(H_1, H_2) \to 2.49139 & (H_2, H_3) \to 3.52491 & (H_3, H_4) \to 2.65238 & (H_4, H_5) \to 3.43621 \\
(H_5, H_6) \to 2.53149 & (H_6, H_7) \to 3.6266 & (H_1, H_3) \to 3.59369 & (H_2, H_4) \to 5.0736 \\
(H_3, H_5) \to 5.26684 & (H_4, H_6) \to 5.27419 & (H_5, H_7) \to 5.15025 & (H_1, H_4) \to 5.28655 \\
(H_2, H_5) \to 5.52935 & (H_3, H_6) \to 7.15038 & (H_4, H_7) \to 5.84691 & (H_1, H_5) \to 6.39121 \\
(H_2, H_6) \to 7.27334 & (H_3, H_7) \to 7.43193 & (H_1, H_6) \to 8.6303 & (H_2, H_7) \to 8.4402 \\
(H_1, H_7) \to 10.0327 &&&
\end{array}
\tag{7.26}
$$

$$
\begin{array}{llll}
(H_1, H_2) \to 2.49139 & (H_2, H_3) \to 3.52491 & (H_3, H_4) \to 2.65238 & (H_4, H_5) \to 3.43621 \\
(H_5, H_6) \to 2.53149 & (H_6, H_7) \to 3.6266 & (H_1, H_3) \to 3.59369 & (H_2, H_4) \to 5.0736 \\
(H_3, H_5) \to 5.26684 & (H_4, H_6) \to 5.27419 & (H_5, H_7) \to 4.59207 & (H_1, H_4) \to 5.28655 \\
(H_2, H_5) \to 5.52935 & (H_3, H_6) \to 7.15038 & (H_4, H_7) \to 5.12442 & (H_1, H_5) \to 6.39121 \\
(H_2, H_6) \to 7.27334 & (H_3, H_7) \to 7.43193 & (H_1, H_6) \to 8.6303 & (H_2, H_7) \to 9.0596 \\
(H_1, H_7) \to 10.0546 &&&
\end{array}
\tag{7.27}
$$

Vertex 96		**Vertex 97**	
Hydrogen Pair	Distance	Hydrogen Pair	Distance
(H_1,H_2)	2.49139	(H_1,H_2)	2.49139
(H_2,H_3)	3.52491	(H_2,H_3)	3.52491
(H_1,H_3)	*3.59369*	*(H_1,H_3)*	*3.59369*
(H_3,H_4)	2.65238	(H_3,H_4)	2.65238
(H_4,H_5)	3.43621	(H_4,H_5)	3.43621
(H_5,H_6)	2.53149	(H_5,H_6)	2.53149
(H_6,H_7)	3.6266	(H_6,H_7)	3.6266
-		*(H_5,H_7)*	*4.59207*

Table 36 – Comparison of distances between solutions 96 and 97 (shown in figure 73). The distances used in the prune are underlined. Distances used for the cutoff value of 4.6 Å are in green and additional distances used for the cutoff value of 4.9 Å are in blue.

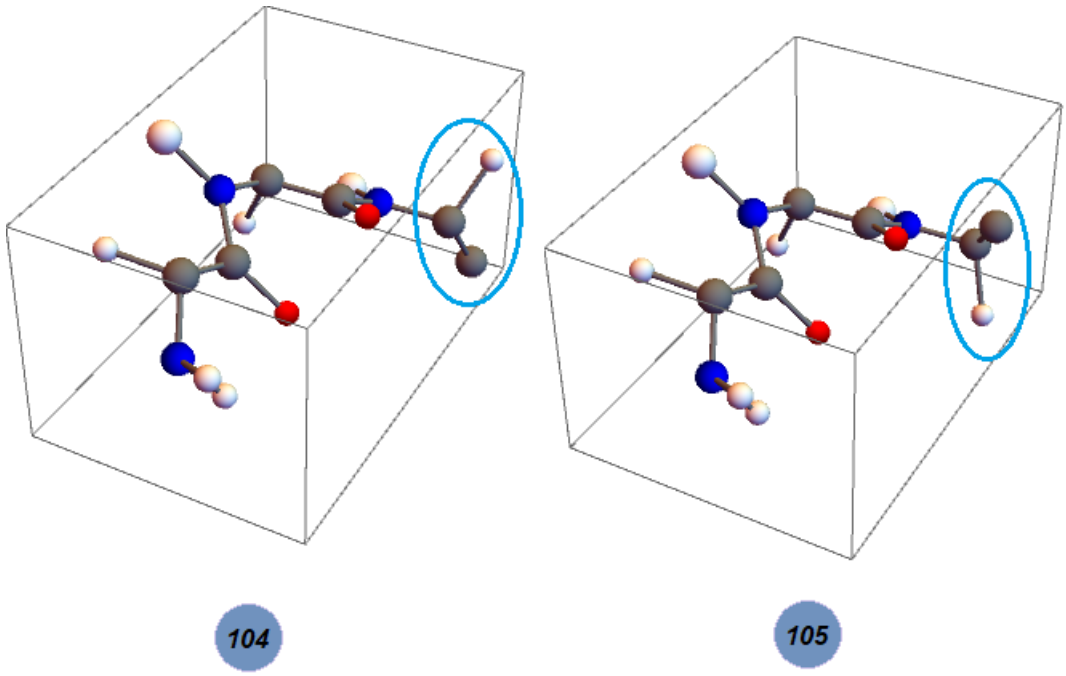

Figure 74 – Solutions for vertices 104 and 105 of the search tree. Distance list for vertex 104 is found in equation 7.28 and distance list for vertex 105 is found in equation 7.29.

Vertex 104		**Vertex 105**	
Hydrogen Pair	Distance	Hydrogen Pair	Distance
(H_1,H_2)	2.49139	(H_1,H_2)	2.49139
(H_2,H_3)	3.52491	(H_2,H_3)	3.52491
(H_1,H_3)	*3.59369*	*(H_1,H_3)*	*3.59369*
(H_3,H_4)	2.53149	(H_3,H_4)	2.53149
(H_4,H_5)	3.6266	(H_4,H_5)	3.6266
(H_2,H_5)	*4.76188*	*(H_2,H_5)*	*4.76188*
(H_3,H_5)	*4.59207*	*(H_3,H_5)*	*4.59207*
(H_5,H_6)	2.65238	(H_5,H_6)	2.65238
(H_6,H_7)	3.43621	(H_6,H_7)	3.43621
-		*(H_5,H_7)*	*4.63118*

Table 37 – Comparison of distances between solutions 104 and 105 (shown in figure 74). The distances used in the prune are underlined. Distances used for the cutoff value of 4.6 Å are in green and additional distances used for the cutoff value of 4.9 Å are in blue.

$$
\begin{array}{llll}
(H_1, H_2) \rightarrow 2.49139 & (H_2, H_3) \rightarrow 3.52491 & (H_3, H_4) \rightarrow 2.53149 & (H_4, H_5) \rightarrow 3.6266 \\
(H_5, H_6) \rightarrow 2.65238 & (H_6, H_7) \rightarrow 3.43621 & (H_1, H_3) \rightarrow 3.59369 & (H_2, H_4) \rightarrow 5.03461 \\
(H_3, H_5) \rightarrow 4.59207 & (H_4, H_6) \rightarrow 5.56355 & (H_5, H_7) \rightarrow 5.26684 & (H_1, H_4) \rightarrow 5.33484 \\
(H_2, H_5) \rightarrow 4.76188 & (H_3, H_6) \rightarrow 6.98832 & (H_4, H_7) \rightarrow 6.82448 & (H_1, H_5) \rightarrow 6.49867 \\
(H_2, H_6) \rightarrow 6.61767 & (H_3, H_7) \rightarrow 8.28216 & (H_1, H_6) \rightarrow 8.30505 & (H_2, H_7) \rightarrow 7.46832 \\
(H_1, H_7) \rightarrow 8.4855 & & &
\end{array}
\tag{7.28}
$$

$$
\begin{array}{llll}
(H_1, H_2) \rightarrow 2.49139 & (H_2, H_3) \rightarrow 3.52491 & (H_3, H_4) \rightarrow 2.53149 & (H_4, H_5) \rightarrow 3.6266 \\
(H_5, H_6) \rightarrow 2.65238 & (H_6, H_7) \rightarrow 3.43621 & (H_1, H_3) \rightarrow 3.59369 & (H_2, H_4) \rightarrow 5.03461 \\
(H_3, H_5) \rightarrow 4.59207 & (H_4, H_6) \rightarrow 5.56355 & (H_5, H_7) \rightarrow 4.63118 & (H_1, H_4) \rightarrow 5.33484 \\
(H_2, H_5) \rightarrow 4.76188 & (H_3, H_6) \rightarrow 6.98832 & (H_4, H_7) \rightarrow 7.04444 & (H_1, H_5) \rightarrow 6.49867 \\
(H_2, H_6) \rightarrow 6.61767 & (H_3, H_7) \rightarrow 7.70401 & (H_1, H_6) \rightarrow 8.30505 & (H_2, H_7) \rightarrow 5.83569 \\
(H_1, H_7) \rightarrow 7.41799 & & &
\end{array}
\tag{7.29}
$$

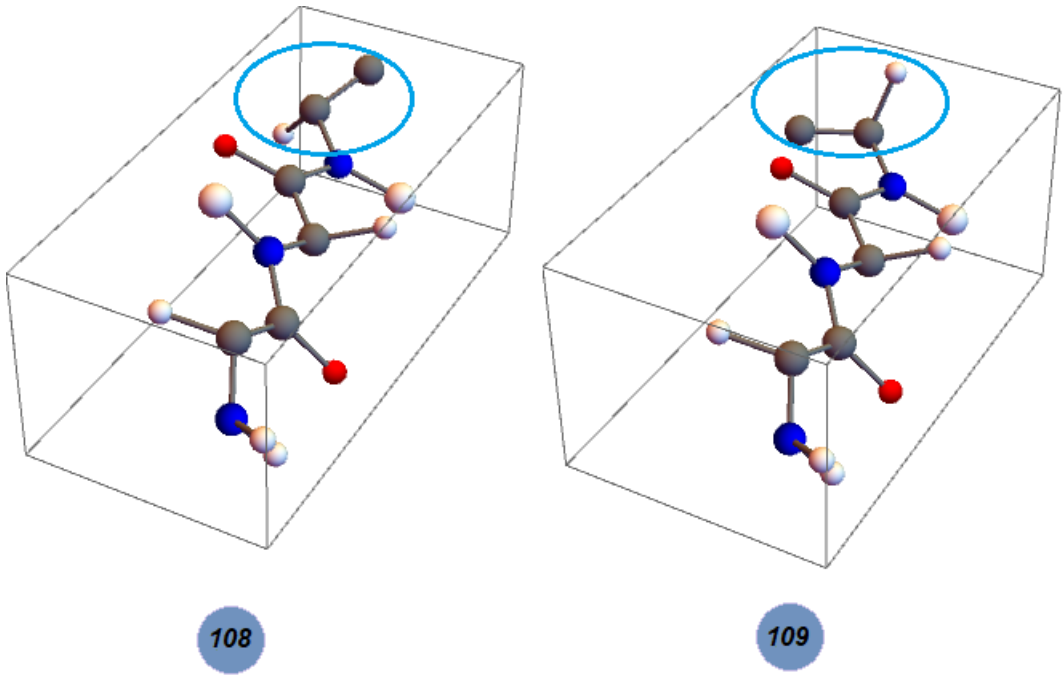

Figure 75 – Solutions for vertices 108 and 109 of the search tree. Distance list for vertex 108 is found in equation 7.30 and distance list for vertex 109 is found in equation 7.31.

$$
\begin{array}{llll}
(H_1, H_2) \to 2.49139 & (H_2, H_3) \to 3.52491 & (H_3, H_4) \to 2.53149 & (H_4, H_5) \to 3.6266 \\
(H_5, H_6) \to 2.65238 & (H_6, H_7) \to 3.43621 & (H_1, H_3) \to 3.59369 & (H_2, H_4) \to 5.03461 \\
(H_3, H_5) \to 5.15025 & (H_4, H_6) \to 5.37946 & (H_5, H_7) \to 5.26684 & (H_1, H_4) \to 5.33484 \\
(H_2, H_5) \to 5.19592 & (H_3, H_6) \to 7.11798 & (H_4, H_7) \to 5.98868 & (H_1, H_5) \to 6.17555 \\
(H_2, H_6) \to 7.12289 & (H_3, H_7) \to 7.88446 & (H_1, H_6) \to 8.58179 & (H_2, H_7) \to 8.93543 \\
(H_1, H_7) \to 10.4408 &&&
\end{array}
\tag{7.30}
$$

$$
\begin{array}{llll}
(H_1, H_2) \to 2.49139 & (H_2, H_3) \to 3.52491 & (H_3, H_4) \to 2.53149 & (H_4, H_5) \to 3.6266 \\
(H_5, H_6) \to 2.65238 & (H_6, H_7) \to 3.43621 & (H_1, H_3) \to 3.59369 & (H_2, H_4) \to 5.03461 \\
(H_3, H_5) \to 5.15025 & (H_4, H_6) \to 5.37946 & (H_5, H_7) \to 4.63118 & (H_1, H_4) \to 5.33484 \\
(H_2, H_5) \to 5.19592 & (H_3, H_6) \to 7.11798 & (H_4, H_7) \to 5.61175 & (H_1, H_5) \to 6.17555 \\
(H_2, H_6) \to 7.12289 & (H_3, H_7) \to 8.01035 & (H_1, H_6) \to 8.58179 & (H_2, H_7) \to 9.34248 \\
(H_1, H_7) \to 10.289 &&&
\end{array}
\tag{7.31}
$$

Vertex 108		**Vertex 109**	
Hydrogen Pair	Distance	Hydrogen Pair	Distance
(H_1,H_2)	2.49139	(H_1,H_2)	2.49139
(H_2,H_3)	3.52491	(H_2,H_3)	3.52491
(H_1,H_3)	*3.59369*	(H_1,H_3)	*3.59369*
(H_3,H_4)	2.53149	(H_3,H_4)	2.53149
(H_4,H_5)	3.6266	(H_4,H_5)	3.6266
(H_5,H_6)	2.65238	(H_5,H_6)	2.65238
(H_6,H_7)	3.43621	(H_6,H_7)	3.43621
-		(H_5,H_7)	*4.63118*

Table 38 – Comparison of distances between solutions 108 and 109 (shown in figure 75). The distances used in the prune are underlined. Distances used for the cutoff value of 4.6 Å are in green and additional distances used for the cutoff value of 4.9 Å are in blue.

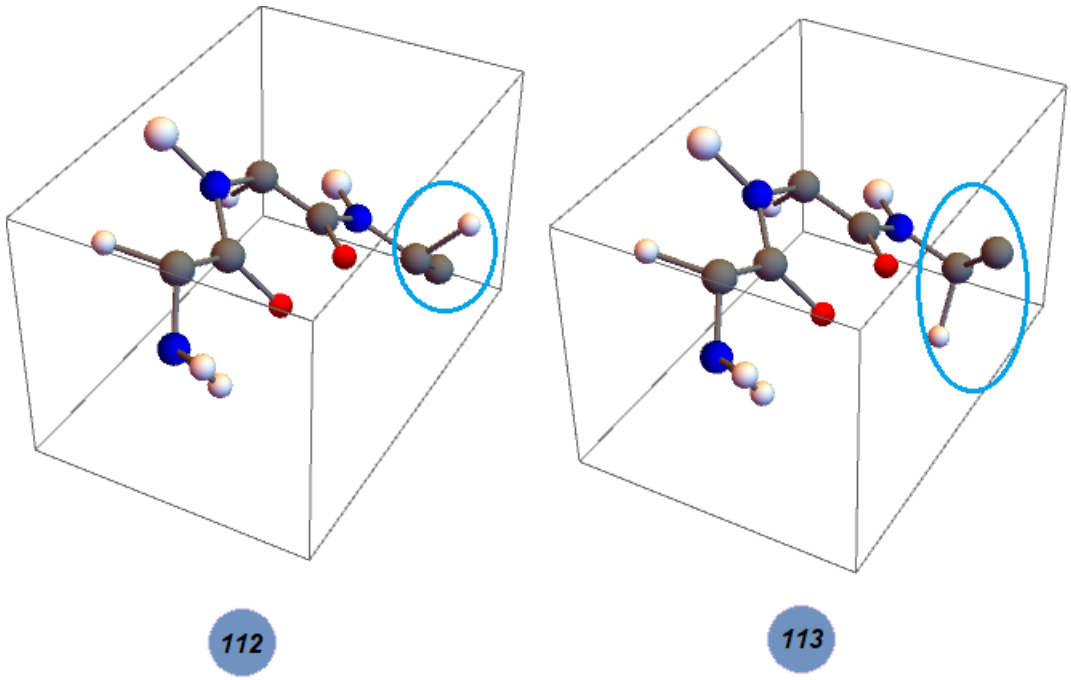

Figure 76 – Solutions for vertices 112 and 113 of the search tree. Distance list for vertex 112 is found in equation 7.32 and distance list for vertex 113 is found in equation 7.33.

Vertex 112		**Vertex 113**	
Hydrogen Pair	Distance	Hydrogen Pair	Distance
(H_1,H_2)	2.49139	(H_1,H_2)	2.49139
(H_2,H_3)	3.52491	(H_2,H_3)	3.52491
(H_1,H_3)	*3.59369*	*(H_1,H_3)*	*3.59369*
(H_3,H_4)	2.53149	(H_3,H_4)	2.53149
(H_4,H_5)	3.43621	(H_4,H_5)	3.43621
(H_3,H_5)	*4.62605*	*(H_3,H_5)*	*4.62605*
(H_5,H_6)	2.65238	(H_5,H_6)	2.65238
(H_6,H_7)	3.6266	(H_6,H_7)	3.6266
-		*(H_5,H_7)*	*4.58485*

Table 39 – Comparison of distances between solutions 112 and 113 (shown in figure 76). The distances used in the prune are underlined. Distances used for the cutoff value of 4.6 Å are in green and additional distances used for the cutoff value of 4.9 Å are in blue.

$$
\begin{array}{llll}
(H_1,H_2) \rightarrow 2.49139 & (H_2,H_3) \rightarrow 3.52491 & (H_3,H_4) \rightarrow 2.53149 & (H_4,H_5) \rightarrow 3.43621 \\
(H_5,H_6) \rightarrow 2.65238 & (H_6,H_7) \rightarrow 3.6266 & (H_1,H_3) \rightarrow 3.59369 & (H_2,H_4) \rightarrow 5.03461 \\
(H_3,H_5) \rightarrow 4.62605 & (H_4,H_6) \rightarrow 5.54055 & (H_5,H_7) \rightarrow 5.20287 & (H_1,H_4) \rightarrow 5.33484 \\
(H_2,H_5) \rightarrow 5.15635 & (H_3,H_6) \rightarrow 6.96997 & (H_4,H_7) \rightarrow 6.74999 & (H_1,H_5) \rightarrow 6.78496 \\
(H_2,H_6) \rightarrow 6.65475 & (H_3,H_7) \rightarrow 7.79182 & (H_1,H_6) \rightarrow 8.36265 & (H_2,H_7) \rightarrow 6.43289 \\
(H_1,H_7) \rightarrow 7.62596 & & &
\end{array}
\tag{7.32}
$$

$$
\begin{array}{llll}
(H_1,H_2) \rightarrow 2.49139 & (H_2,H_3) \rightarrow 3.52491 & (H_3,H_4) \rightarrow 2.53149 & (H_4,H_5) \rightarrow 3.43621 \\
(H_5,H_6) \rightarrow 2.65238 & (H_6,H_7) \rightarrow 3.6266 & (H_1,H_3) \rightarrow 3.59369 & (H_2,H_4) \rightarrow 5.03461 \\
(H_3,H_5) \rightarrow 4.62605 & (H_4,H_6) \rightarrow 5.54055 & (H_5,H_7) \rightarrow 4.58485 & (H_1,H_4) \rightarrow 5.33484 \\
(H_2,H_5) \rightarrow 5.15635 & (H_3,H_6) \rightarrow 6.96997 & (H_4,H_7) \rightarrow 6.82054 & (H_1,H_5) \rightarrow 6.78496 \\
(H_2,H_6) \rightarrow 6.65475 & (H_3,H_7) \rightarrow 7.15323 & (H_1,H_6) \rightarrow 8.36265 & (H_2,H_7) \rightarrow 5.03576 \\
(H_1,H_7) \rightarrow 6.94711 & & &
\end{array}
\tag{7.33}
$$

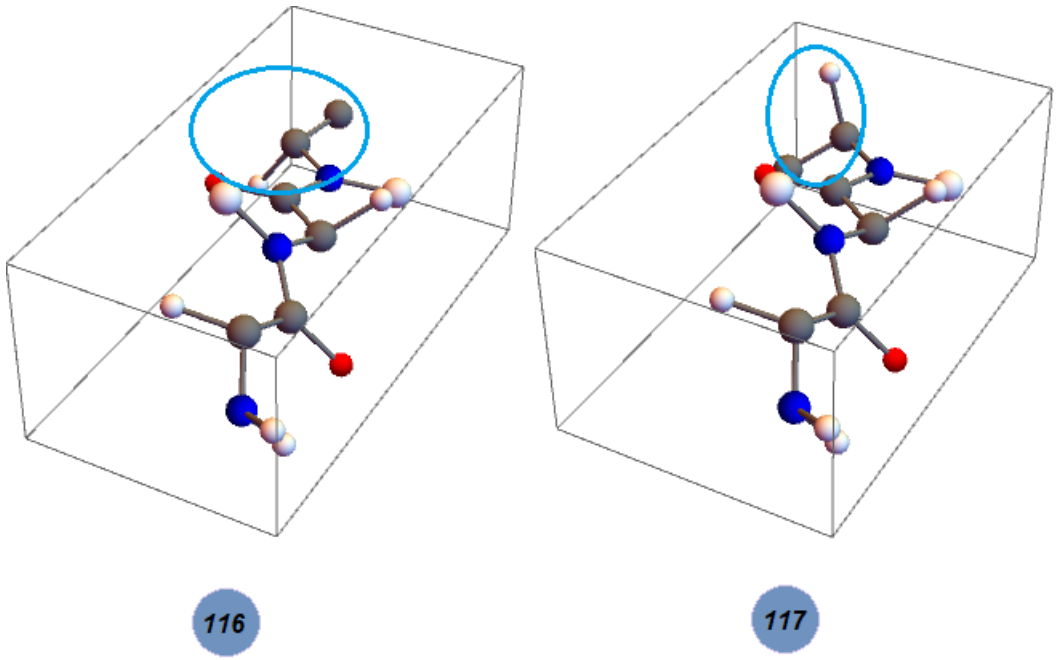

Figure 77 – Solutions for vertices 116 and 117 of the search tree. Distance list for vertex 116 is found in equation 7.34 and distance list for vertex 117 is found in equation 7.35.

$$
\begin{array}{llll}
(H_1, H_2) \to 2.49139 & (H_2, H_3) \to 3.52491 & (H_3, H_4) \to 2.53149 & (H_4, H_5) \to 3.43621 \\
(H_5, H_6) \to 2.65238 & (H_6, H_7) \to 3.6266 & (H_1, H_3) \to 3.59369 & (H_2, H_4) \to 5.03461 \\
(H_3, H_5) \to 5.20159 & (H_4, H_6) \to 5.34811 & (H_5, H_7) \to 5.20287 & (H_1, H_4) \to 5.33484 \\
(H_2, H_5) \to 5.57526 & (H_3, H_6) \to 7.10513 & (H_4, H_7) \to 5.80997 & (H_1, H_5) \to 6.46344 \\
(H_2, H_6) \to 7.1768 & (H_3, H_7) \to 7.33043 & (H_1, H_6) \to 8.64839 & (H_2, H_7) \to 8.24638 \\
(H_1, H_7) \to 9.94343 & & &
\end{array}
\tag{7.34}
$$

$$
\begin{array}{llll}
(H_1, H_2) \to 2.49139 & (H_2, H_3) \to 3.52491 & (H_3, H_4) \to 2.53149 & (H_4, H_5) \to 3.43621 \\
(H_5, H_6) \to 2.65238 & (H_6, H_7) \to 3.6266 & (H_1, H_3) \to 3.59369 & (H_2, H_4) \to 5.03461 \\
(H_3, H_5) \to 5.20159 & (H_4, H_6) \to 5.34811 & (H_5, H_7) \to 4.58485 & (H_1, H_4) \to 5.33484 \\
(H_2, H_5) \to 5.57526 & (H_3, H_6) \to 7.10513 & (H_4, H_7) \to 5.22397 & (H_1, H_5) \to 6.46344 \\
(H_2, H_6) \to 7.1768 & (H_3, H_7) \to 7.44546 & (H_1, H_6) \to 8.64839 & (H_2, H_7) \to 9.00778 \\
(H_1, H_7) \to 10.1394 & & &
\end{array}
\tag{7.35}
$$

Vertex 116		**Vertex 117**	
Hydrogen Pair	Distance	Hydrogen Pair	Distance
(H_1,H_2)	2.49139	(H_1,H_2)	2.49139
(H_2,H_3)	3.52491	(H_2,H_3)	3.52491
(H_1,H_3)	*3.59369*	*(H_1,H_3)*	*3.59369*
(H_3,H_4)	2.53149	(H_3,H_4)	2.53149
(H_4,H_5)	3.43621	(H_4,H_5)	3.43621
(H_5,H_6)	2.65238	(H_5,H_6)	2.65238
(H_6,H_7)	3.6266	(H_6,H_7)	3.6266
-		*(H_5,H_7)*	*4.58485*

Table 40 – Comparison of distances between solutions 116 and 117 (shown in figure 77). The distances used in the prune are underlined. Distances used for the cutoff value of 4.6 Å are in green and additional distances used for the cutoff value of 4.9 Å are in blue.

7.4 Computational Results

In this section the results regarding computational times for the modified Branch-and-Prune algorithm to find its first solution will be given. These tests were made for instances having 3, 5, 8, 10, 12, 15, 20, 25, 30 and 40 amino acids and cuts of 4.5 Å, 5 Å, 5.5 Å, 6 Å, 6.5 Å and no cuts. For each cut/size combination 10 tests were made with randomly generated samples and the average computation time was calculated. All tests were run in Mathematica, using a Aspire F5-573 computer with Intel core $i5$ 7^{th} generation processor and 8 GB of RAM.

# Amino acids	3	5	8	10	12	15	20	25	30	40
Time (in seconds)	0.36	1.03	3.30	5.71	9.01	15.96	47.58	63.44	85.46	190.67

Table 41 – Execution time for instances having all hydrogen pair distances (no cuts)

# Amino acids	3	5	8	10	12	15	20	25	30	40
Time (in seconds)	0.27	0.81	2.29	3.89	5.16	10.86	16.26	29.37	78.98	85.09

Table 42 – Execution time for instances having a cut of 4.5Å

# Amino acids	3	5	8	10	12	15	20	25	30	40
Time (in seconds)	0.28	0.88	2.48	4.07	5.49	8.79	15.48	26.96	35.19	75.42

Table 43 – Execution time for instances having a cut of 5Å

The fact that the computational time in all cut cases is very similar is rather surprising. However, the cases not having all distances were tested already knowing what the solution was, and this test was halted as soon as this solution was found. This way, the test didn't necessarily stopped when the first solution was found, but it didn't necessarily run until all solutions were found and the whole algorithm decision tree was spanned.

# Amino acids	3	5	8	10	12	15	20	25	30	40
Time (in seconds)	0.26	0.80	2.36	3.41	7.29	9.38	17.87	26.90	49.05	85.19

Table 44 – Execution time for instances having a cut of 5.5Å

# Amino acids	3	5	8	10	12	15	20	25	30	40
Time (in seconds)	0.30	0.88	2.27	4.06	6.45	9.96	17.14	29.43	40.84	80.88

Table 45 – Execution time for instances having a cut of 6Å

# Amino acids	3	5	8	10	12	15	20	25	30	40
Time (in seconds)	0.26	0.85	2.76	4.76	5.99	10.10	17.83	40.28	41.35	91.36

Table 46 – Execution time for instances having a cut of 6.5Å

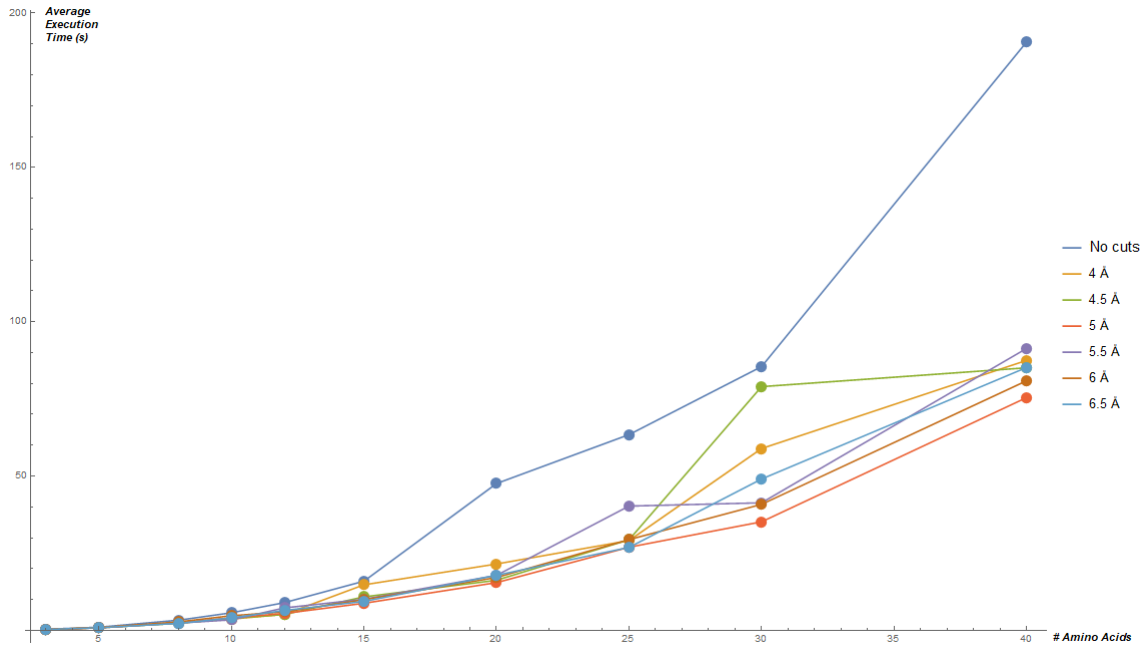

Figure 78 – Comparison between execution times with different cutoff values and sizes.

Since the cases having cuts (ie, where not all distances are provided) have less prunes, small instances in these tests tend to run on a smaller computational time. This also happens due to the fact that, the smaller the instance, the smaller the cardinality of the solution set is. Hence, the right solution tends to be found more quickly.

In the cases where all available distances are provided it can be considered that the computational times shown are the actual algorithmic computational time. As was previously said for the cases having cuts, these tests also were programmed to halt as soon as the correct solution was found, but from chapter 6 it is known that, when all distances are provided, the solution for this problem is unique.

These cases having no cuts also take longer in average. One reason for this could be that, with all distances provided, more restrictions are given and hence the

situation where the algorithm descends in a branch until a node next to a leaf but a prune occurs probably happens more often, augmenting the execution time. On the other hand, for practical situations (were the right solution is not known), the fact that the solution is unique makes these cases more reliable in terms of correct solutions.

In figure 78 all data regarding these tests is put into a single graph so that all combinations of sizes and cutoff values can better be visualized and compared. It can be clearly seen that the computation time seems to increase exponentially, proportional to the number of amino acids in the protein.

For the cases where there are distance cuts it takes too long to find all solutions in order for them to be compared. The tests made with an instance with size 3 took more than 5 minutes to be completed and in the case of an instance of size 5, after computing the results for a whole night, it still hadn't found all solutions. It is imperative, therefore, that we look for alternatives to solve this specific problem.

8 Conclusion

In chapter 1 a description of the structures of proteins and how they are used in this work was presented. Along that, the concept of graphs - and more specifically, the graph used to model these proteins for the Unassigned Distance Geometry Problem - was introduced. This introductory chapter laid part of the foundation for the concepts that permeated the entire work, such as the disregard for the proteins' side chains (and why this is a valid action to be taken) and the variables used in the protein model:

- r_{v_i,v_j}: distance between atoms v_i and v_j having a covalent bond (equal to 1.526 Å)
- θ_{v_i,v_k}: angle between three atoms v_i, v_j, v_k where v_i, v_j and v_j, v_k have a covalent bond (1.91 rad)
- ω_{v_i,v_l}: angle between the planes formed by the atoms v_i, v_j, v_k e v_j, v_k, v_l where v_i and v_l are separated by three covalent bonds
- x_{v_i}: spatial coordinates of atom v_i

Chapter 2 introduced important definitions: the Distance Geometry Problem, concepts of rigidity and its connection to this problem and its Assigned and Unassigned classes, explaining why this work's category falls into the Unassigned class. It also brings a general historical view of the progressions in the area and important primary results, such as the problem being NP-Hard.

In chapter 3 vertex orders were presented. This concept has ties to both theoretical concepts (such as the Discretizable Distance Geometry Problem and the fact that all instances for this class can be ordered according to definition 12) and more practical concepts as well, like its important role in the Branch-and-Prune algorithm.

In chapter 4 the process of instance generation was explained, in order to lay the path for the introduction of instance generation in this work (presented in a chapter ahead). As with the Branch-and-Prune algorithm, the process of instance generation used here is a modification from what can be seen in the Lavor Instances. Also in line with the modifications made in the Branch-and-Prune algorithm, this part of the work is based in what is described in section 4.1 and takes advantage of the rigid substructures in the molecule's models for speeding up the process.

The original Branch-and-Prune algorithm was presented in chapter 5 and its main procedures were explained, which ties the previous chapters and shows how each of the definitions presented before are applied to the resolution of the Distance Geometry

Problem. It starts with an instance such as it was described in chapter 4 whose distances are used to realize the proteins' coordinated according to the hand-crafted vertex order (chapter 3). This algorithm was conceived as a way to find a solution for the associated Distance Geometry Problem, and is the foundation for the modified algorithm described ahead that aims to resolve the Unassigned Distance Geometry Problem.

Chapter 7 takes the theory developed in the previous chapters and applies it to examples to better illustrate the results of this work. Firstly, a basic example with the first iterations of the modified branch-and-prune algorithm is shown with no cutoff value for the instance distances. Following this, a more complete example of the modified branch-and-prune algorithm is fully described, including all of its iterations.

After these two examples of the algorithm's iterations an analysis of the algorithm search tree's variations is conducted with three instances generated from the same solution and varying cutoff values. Finally, in the last section of this chapter a comparison of different cutoff values and molecule sizes is made in line with the assumption that the algorithm's execution time is exponentially proportional to its instance's size, which is, in turn, related to its' proteins' sizes (that is, the size of the solutions looked for the algorithm).

Results regarding the use of symmetries in the original Branch-and-Prune algorithm in order to diminish the computation time and find all solutions of a given instance are already available in the literature (as seen in references (MUCHERINO C. LAVOR, 2012) and (LAVOR et al., 2017)). Given the fact that both the MDGP and the DMDGP are NP-hard problems (reference (MUCHERINO C. LAVOR, 2012)), the computation time can increase exponentially with the increase of the instance's size, and this continues to be the case for the Unassigned problems discussed in this work.

Indeed, not only the problem continues to be exponential (given the fact that the original BP tree is a subtree in the modified BP algorithm), but it can have more solutions to be found - making the calculations even longer. With this in mind, a valid path for following works is, starting with the ideas exploring symmetries in the original BP algorithm, identifying ways to make faster calculations to find all solutions for instances of the Unassigned case.

Observations about the possible symmetries for the Unassigned DMDGP were made in chapter 7, at the same time that it was observed that they do not appear always (depending on the cutoff value set in the problem). Therefore, using symmetries in this case would have to be preceded by a more in-depth analysis of cases where the cutoff value could imppose a complication for the full symmetry to appear.

Nevertheless, some symmetry does seem to exist and a possible way to tackle cases where the cutoff value is an impediment for a fully symmetric tree would be to

perform a post-check in all distances for the solutions found and compare them to the instance's distance list, excluding those solutions whose distances don't match.

Another way to make calculations faster is to extend the approach made in this work using rigid substructures, attempting to incorporate larger structures at each iteration of the algorithm. A possibility for this would be, for instance, incorporating a full amino acid instead of half at each stage.

Bibliography

A VALADARES NF, N. d. S. O. e. a. B. *Protein Structure, Modelling and Applications*. Bethesda (MD): National Center for Biotechnology Information (US), 2006. (Gruber A, Durham AM, Huynh C, et al., editors. Bioinformatics in Tropical Disease Research: A Practical and Case-Study Approach). Disponível em: <https://www.ncbi.nlm.nih.gov/books/NBK6824/>. Citado na página 19.

ALVES, R.; LAVOR, C. Geometric algebra to model uncertainties in the discretizable molecular distance geometry problem. *Advances in Applied Clifford Algebras*, v. 27, n. 1, p. 439–452, Mar 2017. ISSN 1661-4909. Disponível em: <https://doi.org/10.1007/s00006-016-0653-2>. Citado na página 33.

BAJAJ, C. The algebraic degree of geometric optimization problems. *Discrete & Computational Geometry*, v. 3, n. 2, p. 177–191, Jun 1988. ISSN 1432-0444. Disponível em: <https://doi.org/10.1007/BF02187906>. Citado na página 27.

BENEDETTI, R.; RISLER, J. *Real algebraic and semi-algebraic sets*. Hermann, 1990. (Actualités mathématiques). ISBN 9782705661441. Disponível em: <https://books.google.com.br/books?id=6___uAAAAMAAJ>. Citado na página 28.

BLUMENTHAL, L. *Theory and applications of distance geometry*. Chelsea Pub. Co., 1970. ISBN 9780828402422. Disponível em: <https://books.google.com.br/books?id=QdcPAQAAMAAJ>. Citado na página 26.

BODLAENDER, H. L.; FOMIN, F. V.; KOSTER, A. M. C. A.; KRATSCH, D.; THILIKOS, D. M. A note on exact algorithms for vertex ordering problems on graphs. *Theory of Computing Systems*, v. 50, n. 3, p. 420–432, Apr 2012. ISSN 1433-0490. Disponível em: <https://doi.org/10.1007/s00224-011-9312-0>. Citado na página 31.

BROWN, A. C. On the theory of isomeric compounds. *J. Chem. Soc.*, The Royal Society of Chemistry, v. 18, p. 230–245, 1865. Disponível em: <http://dx.doi.org/10.1039/JS8651800230>. Citado na página 22.

CONNELLY, B. Chapter 2: Basic concepts. In: . [s.n.], 1987. Disponível em: <http://www.math.cornell.edu/~connelly/rigidity.chapter.2.pdf>. Citado na página 28.

CRIPPEN, G.; HAVEL, T. Distance geometry and molecular conformation. v. 15, 01 1988. Citado 3 vezes nas páginas 16, 26, and 51.

DONALD, B. R. *Algorithms in Structural Molecular Biology*. [S.l.]: The MIT Press, 2011. ISBN 0262015595, 9780262015592. Citado 2 vezes nas páginas 28 and 51.

DUXBURY L. GRANLUND, S. G. P. J. S. B. P. The unassigned distance geometry problem. *Discrete Applied Mathematics*, v. 204, p. 117 – 132, 2016. ISSN 0166-218X. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0166218X15005168>. Citado na página 29.

DUXBURY, P.; LAVOR, C.; LIBERTI, L.; SALLES-NETO, L. L. de. Unassigned distance geometry and molecular conformation problems. *Journal of Global Optimization*,

2021. Disponível em: <http://www.lix.polytechnique.fr/Labo/Leo.Liberti/jogo21d.pdf>. Citado na página 29.

EDUCATION, M. F. for M.; (MFMER), R. How prions fold. Disponível em: <https://www.mayoclinic.org/diseases-conditions/creutzfeldt-jakob-disease/multimedia/normal-and-diseased-prions/img-20007478>. Citado 2 vezes nas páginas 9 and 16.

FIDALGO, F.; GONÇALVES, D. S.; LAVOR, C.; LIBERTI, L.; MUCHERINO, A. A symmetry-based splitting strategy for discretizable distance geometry problems. *Journal of Global Optimization*, v. 71, n. 4, p. 717–733, Aug 2018. ISSN 1573-2916. Disponível em: <https://doi.org/10.1007/s10898-018-0610-9>. Citado na página 44.

FLORENCIO, A. *Para que serve a quiralidade?* 2016. Disponível em: <https://universidadedaquimica.com.br/wp-content/uploads/2016/01/diasteroisomeros-825x444.jpg>. Citado 2 vezes nas páginas 10 and 53.

GIBSON, K. D.; SCHERAGA, H. A. Energy minimization of rigid-geometry polypeptides with exactly closed disulfide loops. v. 18, p. 403 – 415, 02 1997. Citado 2 vezes nas páginas 22 and 31.

GONÇALVES, D. S.; MUCHERINO, A. Discretization orders and efficient computation of cartesian coordinates for distance geometry. *Optimization Letters*, v. 8, n. 7, p. 2111–2125, Oct 2014. ISSN 1862-4480. Disponível em: <https://doi.org/10.1007/s11590-014-0724-z>. Citado na página 32.

GRAVER, J.; AMERICA, M. A. of. *Counting on Frameworks: Mathematics to Aid the Design of Rigid Structures*. Mathematical Association of America, 2001. (Dolciani Mathematical Expositions). ISBN 9780883853313. Disponível em: <https://books.google.com.br/books?id=0MCpQx5wJ74C>. Citado 2 vezes nas páginas 27 and 28.

HENDRICKSON, B. Conditions for unique graph realizations. *SIAM Journal on Computing*, v. 21, n. 1, p. 65–84, 1992. Disponível em: <https://doi.org/10.1137/0221008>. Citado na página 31.

_____. The molecule problem: Exploiting structure in global optimization. *SIAM Journal on Optimization*, v. 5, n. 4, p. 835–857, 1995. Disponível em: <https://doi.org/10.1137/0805040>. Citado 5 vezes nas páginas 7, 8, 18, 48, and 50.

HENNEBERG, L. *Statik der starren Systeme*. A. Bergstraesser, 1886. (Lehrbuch der technischen Mechanik). Disponível em: <https://books.google.com.br/books?id=z4pNAAAAYAAJ>. Citado na página 31.

JACKSON, B.; JORDÁN, T. On the rigidity of molecular graphs. *Combinatorica*, v. 28, n. 6, p. 645–658, Nov 2008. ISSN 1439-6912. Disponível em: <https://doi.org/10.1007/s00493-008-2287-z>. Citado na página 29.

LAMAN, G. On graphs and rigidity of plane skeletal structures. *Journal of Engineering Mathematics*, v. 4, n. 4, p. 331–340, Oct 1970. ISSN 1573-2703. Disponível em: <https://doi.org/10.1007/BF01534980>. Citado na página 29.

LANSBURY, P. T.; CAUGHEY, B. The double life of the prion protein. *Current Biology*, v. 6, n. 8, p. 914 – 916, 1996. ISSN 0960-9822. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0960982202006243>. Citado na página 16.

LAVOR, C. On generating instances for the molecular distance geometry problem. In: _____. *Global Optimization: From Theory to Implementation*. Boston, MA: Springer US, 2006. p. 405–414. Disponível em: <https://doi.org/10.1007/0-387-30528-9_14>. Citado 3 vezes nas páginas 9, 39, and 41.

LAVOR, C.; LIBERTI, L.; DONALD, B.; WORLEY, B.; BARDIAUX, B.; MALLIAVIN, T. E.; NILGES, M. Minimal nmr distance information for rigidity of protein graphs. *Discrete Applied Mathematics*, 2018. ISSN 0166-218X. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0166218X18301793>. Citado 19 vezes nas páginas 7, 8, 9, 16, 17, 18, 20, 21, 22, 23, 26, 27, 31, 32, 33, 36, 37, 38, and 44.

LAVOR, C.; LIBERTI, L.; MACULAN, N. Computational experience with the molecular distance geometry problem. In: _____. *Global Optimization: Scientific and Engineering Case Studies*. Boston, MA: Springer US, 2006. p. 213–225. ISBN 978-0-387-30927-9. Disponível em: <https://doi.org/10.1007/0-387-30927-6_9>. Citado na página 27.

LAVOR, C.; LIBERTI, L.; MACULAN, N.; MUCHERINO, A. The discretizable molecular distance geometry problem. *Computational Optimization and Applications*, v. 52, n. 1, p. 115–146, May 2012. ISSN 1573-2894. Disponível em: <https://doi.org/10.1007/s10589-011-9402-6>. Citado 2 vezes nas páginas 31 and 32.

LAVOR, C.; LIBERTI, L.; MUCHERINO, A. The interval branch-and-prune algorithm for the discretizable molecular distance geometry problem with inexact distances. *Journal of Global Optimization*, v. 56, n. 3, p. 855–871, Jul 2013. ISSN 1573-2916. Disponível em: <https://doi.org/10.1007/s10898-011-9799-6>. Citado 9 vezes nas páginas 9, 19, 23, 24, 25, 32, 33, 34, and 35.

LAVOR, C.; MACULAN, N.; SOUZA, M.; ALVES, R. *Theory and applications of distance geometry*. 2017. Citado na página 118.

LAVOR, C.; MUCHERINO, A.; LIBERTI, L.; MACULAN, N. *The Discretizable Molecular Distance Geometry Problem*. 2012. Disponível em: <http://www.inf.ufrgs.br/elavio2012/elavio2012/Downloads_files/slides_elavio_carlile_1.pdf>. Citado 2 vezes nas páginas 9 and 17.

LAVOR, L. L. C. *Um Convite à Geometria de Distâncias*. [S.l.]: SBMAC, 2014. v. 71. (Notas em Matemática Aplicada, v. 71). ISBN 978-85-8215-050-4. Citado 3 vezes nas páginas 9, 45, and 46.

LIBERTI, L.; LAVOR, C.; MACULAN, N.; MUCHERINO, A. Euclidean distance geometry and applications. *SIAM Review*, SIAM, v. 56, p. 3–69, 2014. Citado 5 vezes nas páginas 26, 27, 28, 29, and 31.

LIBERTI, L.; MASSON, B.; LEE, J.; LAVOR, C.; MUCHERINO, A. On the number of realizations of certain henneberg graphs arising in protein conformation. *Discrete Applied Mathematics*, v. 165, p. 213 – 232, 2014. ISSN 0166-218X. 10th Cologne/Twente Workshop on Graphs and Combinatorial Optimization (CTW 2011). Disponível em:

<http://www.sciencedirect.com/science/article/pii/S0166218X13000449>. Citado 2 vezes nas páginas 44 and 66.

MENGER, K. Unterschungen über allgemeine metrik. v. 100, p. 75–163, 12 1928. Citado na página 26.

MUCHERINO, A.; LAVOR, C.; LIBERTI, N. M. L. Strategies for solving distance geometry problems with inexact distances by discrete approaches. In: _____. *Proceedings of Toulouse Global Optimization 2010 (TOGO10)*. [S.l.: s.n.], 2010. p. 93–96. Citado na página 16.

MUCHERINO C. LAVOR, L. L. A. Exploiting symmetry properties of the discretizable molecular distance geometry problem. *Journal of Bioinformatics and Computational Biology*, v. 10, n. 3, 2012. Citado 2 vezes nas páginas 84 and 118.

MUELLER, C.; MARTIN, B.; LUMSDAINE, A. A comparison of vertex ordering algorithms for large graph visualization. In: *2007 6th International Asia-Pacific Symposium on Visualization*. [S.l.: s.n.], 2007. p. 141–148. Citado na página 31.

PHILLIPS, A. T.; ROSEN, J. B.; WALKE, V. H. Molecular structure determination by convex global underestimation of local energy minima. In: *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*. [S.l.]: American Mathematical Society, 1994. p. 181–198. Citado na página 39.

R., A.; C., L.; C., S.; M., S. Clifford algebra and discretizable distance geometry. *Mathematical Methods in the Applied Sciences*, v. 0, n. 0, 2017. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1002/mma.4422>. Citado na página 33.

SAXE, J. *Embeddability of Weighted Graphs in K-space is Strongly NP-hard*. Carnegie-Mellon University, Department of Computer Science, 1980. (CMU-CS-80-102). Disponível em: <https://books.google.com.br/books?id=vClAGwAACAAJ>. Citado na página 26.

SCIENCE, G. *Molecular Biology of the Cell, 5th Ed, 2008: Cell*. [s.n.], 2008. (Molecular Biology of the Cell). Disponível em: <https://books.google.com.br/books?id=4ARfDwAAQBAJ>. Citado na página 21.

UNCCH. *Protein Structure and Function*. 2018. Disponível em: <https://biophysics.unc.edu/faculty/protein-structure-and-function/>. Citado 2 vezes nas páginas 9 and 17.

WEST, D. B. *Introduction to Graph Theory*. Second. Upper Saddle River, N.J.: Prentice Hall, 2001. ISBN 0130144002 9780130144003. Citado 2 vezes nas páginas 22 and 65.

YEMINI, Y. *DTIC ADP003801: The Positioning Problem - A Draft of an Intermediate Summary*. [s.n.], 1978. Disponível em: <https://archive.org/stream/DTIC_ADP003801/DTIC_ADP003801_djvu.txt>. Citado na página 26.