ALGORITMOS SEQUENCIAIS E PARALELOS PARA PROBLEMAS DE GEOMETRIA MOLECULAR

Warley Gramacho da Silva

Tese de Doutorado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia de Sistemas e Computação.

Orientadores: Nelson Maculan Filho
Carlile Campos Lavor

Rio de Janeiro
Julho de 2013

ALGORITMOS SEQUENCIAIS E PARALELOS PARA PROBLEMAS DE GEOMETRIA MOLECULAR

Warley Gramacho da Silva

TESE SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Examinada por:

Prof. Nelson Maculan Filho, D.Habil.

Prof. Carlile Campos Lavor, D.Sc.

Prof. Felipe Maia Galvão França, Ph.D.

Prof. Luiz Satoru Ochi, D.Sc.

Prof. Luidi Gelabert Simonetti, D.Sc.

RIO DE JANEIRO, RJ – BRASIL
JULHO DE 2013

Silva, Warley Gramacho da

Algoritmos Sequenciais e Paralelos para Problemas de Geometria Molecular/Warley Gramacho da Silva. – Rio de Janeiro: UFRJ/COPPE, 2013.

XIV, 65 p.: il.; 29, 7cm.

Orientadores: Nelson Maculan Filho

Carlile Campos Lavor

Tese (doutorado) – UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2013.

Referências Bibliográficas: p. 60 – 65.

1. Discretizable Distance Geometry Problem. 2. Branch-and-Prune Algorithm. 3. Proteins. 4. Discretizable Molecular Distance Geometry Problem. I. Maculan Filho, Nelson *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

God, give me grace to accept with serenity the things that cannot be changed, courage to change the things which should be changed, and the wisdom to distinguish the one from the other.

Adaptado de Reinhold Niebuhr

Aos meus pais, Zilda Gramacho da Silva e Joaquim Gramacho da Silva, que dedicaram suas vidas para me dar oportunidades as quais nunca tiveram

Agradecimentos

Primeiramente, agradeço a Deus, por tudo que me é proporcionado todos os dias e por ter permitido a conclusão desta tese.

Agradeço aos meus pais Zilda e Joaquim por todo o amor e carinho, pelo apoio incondicional em todas as fases da minha vida. Aos meus irmãos, Glenda, Wandrey e Wesley que muitas vezes, mesmo não percebendo, tiveram influência no incentivo 'a busca do conhecimento e na minha formação como pessoa.

Á minha querida esposa Glêndara, pelo seu apoio absoluto, amor e paciência em todos os momentos durante a realização desta tese. Nada disso seria possível sem seu apoio. Agradeço também ao sr. Silvio, dona Joana e ao Gabriel pelo apoio e pelos momentos de descontração.

Ao Prof. Nelson Maculan, que em sua infinita experiência me transmitiu ensinamentos que vão além da academia e da profissão, são exemplos de vida!

Ao Prof. Carlile Lavor, que mais uma vez exerceu um papel importante em minha formação acadêmica, atuando como incentivador e motivador de boa parte da minha trajetória profissional.

Ao Prof. Antonio Mucherino, que me acompanhou, juntamente com os meus orientadores, no desenvolvimento deste trabalho.

Agradeço aos membros da banca examinadora: Luiz Satoru, Felipe França e Luidi Simonetti, pelas contribuições.

Um agradecimento a todos os novos amigos que fiz na COPPE e aos velhos colegas e amigos que me acompanharam durante esse processo: Marcelo Lins, Rogério Tostas, Rodrigo (pit), Gentil Veloso, Rogério Azevedo, Hellena Apolinário, Sandra Regina, Ary Henrique, Rafael Lima e Genilson Gama (primo do).

Agradeço ao colegiado de Ciência da Computação da Universidade Federal do Tocantins pelo apoio em algumas etapas desse processo de doutoramento.

Agradeço ao PESC/COPPE e seus professores na pessoa do prof. Adilson Elias Xavier pela ampliação de meus conhecimentos nas diversas áreas do saber.

Peço desculpas às pessoas cujos nomes deveriam estar aqui, mas acabaram ficando de fora.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

ALGORITMOS SEQUENCIAIS E PARALELOS PARA PROBLEMAS DE GEOMETRIA MOLECULAR

Warley Gramacho da Silva

Julho/2013

Orientadores: Nelson Maculan Filho
Carlile Campos Lavor

Programa: Engenharia de Sistemas e Computação

Neste trabalho, propomos uma versão paralela do algoritmo *Branch & Prune* (BP) para o *Discretizable Distance Geometry Problem* (DDGP), que consiste em uma subclasse do *Distance Geometry Problem* (DGP) que pode ser discretizada. A idéia principal é dividir uma instância DDGP em sub-instâncias tantas quanto o número de processos envolvidos na computação paralela e chamar a versão sequencial do BP em cada processo. Devido à flexibilidade de discretização de instâncias DDGP, a subdivisão da instância original pode ser realizada de modo que todas as soluções geradas, para todas as sub-instâncias, são representadas em um sistema de coordenadas comum. Desta forma, a fase de comunicação do algoritmo paralelo, onde as soluções locais são combinadas para gerar o conjunto final de soluções, é muito eficiente. Apresentamos alguns experimentos computacionais, usando proteínas, e estudamos o comportamento do algoritmo em relação ao número de processos considerados. Para o DDGP relacionado a proteínas, o *Discretizable Molecular Distance Geometry Problem* (DMDGP), utilizando distâncias intervalares, conhecido como *Interval Discretizable Molecular Distance Geometry Problem* *i*DMDGP, propomos uma nova ordem para os átomos da cadeia principal de uma molécula de proteína, que permite a aplicação do *Interval Branch & Prune* (*i*BP) que resolve instâncias do *i*DMDGP. Por fim, propomos também um novo algoritmo para o *Discretizing Vertex Order Problem* (DVOP), que é uma importante etapa de pré-processamento do DDGP. Apresentamos alguns resultados computacionais que mostram que o novo algoritmo resolve de forma eficiente o DVOP.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

SEQUENTIAL AND PARALLEL ALGORITHMS FOR MOLECULAR GEOMETRY PROBLEMS

Warley Gramacho da Silva

July/2013

Advisors: Nelson Maculan Filho
Carlile Campos Lavor

Department: Systems Engineering and Computer Science

In this work, we propose a parallel version of the Branch & Prune (BP) algorithm for the Discretizable Distance Geometry Problem (DDGP), which consists in a subclass of Distance Geometry Problems (DGPs) that can be discretized. The main idea is to split a DDGP instance in as many subinstances as the number of processors involved in the computation, and to invoke the sequential version of BP on each processor. Due to the flexibility of the discretization of DDGP instances, the subdivision of the original instance can be performed so that all solutions generated locally solving the several subinstances are represented in a common coordinate system. This way, the communication phase of the parallel algorithm, where the local solutions are combined in order to generate the final set of solutions, is very efficient. We present some computational experiments, using proteins, and study the behavior of the algorithm in relation to the number of considered processors. For DDGP related proteins, The Discretizable Malecular Distance Geometry Problem (DMDGP) using interval distances, known as Interval Discretizable Molecular Distance Geometry Problem (iDMDGP) we propose a new handcrafted order for the protein backbones, which allows the application of the Interval Branch & Prune (iBP) algorithm that resolves instances of iDMDGP. Finally, we also propose a new algorithm for the Discretizing Vertex Order Problem (DVOP), which is an important pre-processing step for the solution of DDGP. We present some computational results showing that the new algorithm efficiently solves the DVOP.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas

iBP	Interval Branch & Prune, p. 3
iDMDGP	Interval Discretizable Molecular Distance Geometry Problem, p. 3
BP	Branch & Prune, p. 2
DDGP	Discretizable Distance Geometry Problem, p. 2
DGP	Distance Geometry Problem, p. 1
DMDGP	Discretizable Molecular Distance Geometry Problem, p. 2
DVOP	Discretizing Vertex Order Problem, p. 2
PDB	Protein Data Bank, p. 1
RMN	Ressonância Magnética Nuclear, p. 1
SNLP	Sensor Network Localization Problem, p. 3

Capítulo 1

Introdução

Um problema bastante estudado pela comunidade científica, que vem aumentando seu número de aplicações na vida real nos últimos anos, é o Problema de Geometria de Distâncias (DGP, isto é, *Distance Geometry Problem*) que consiste em encontrar as coordenadas de um determinado conjunto de pontos, a partir de algumas distâncias conhecidas, entre pares desses pontos [4, 12, 39, 44]. Aplicações muito interessantes são encontradas em cálculos de estrutura molecular [2, 12, 18, 35, 45] e na localização em redes de sensores [3, 10, 19], além do reconhecimento de imagens [26], visualização de informação [15, 50], tomografia da internet [8] ou reconstrução de mapas [13]. Mais recentemente, esta teoria tem sido aplicada no reconhecimento de face [25] e segmentação de imagem [58]. Um recente e detalhado levantamento sobre aplicações do DGP pode ser encontrado em [40].

Em se tratando do DGP aplicado à conformação molecular, informações de algumas distâncias entre pares de átomos que formam uma molécula podem ser fornecidas através de experimentos de Ressonância Magnética Nuclear (RMN). A conformação tridimensional de uma molécula, ou seja, as coordenadas de todos os seus átomos, pode ser determinada através da resolução de um DGP. O DGP relacionado à moléculas é normalmente referido como Problema de Geometria de Distancias Moleculares (em inglês, *Molecular Distance Geometry Problem* - MDGP).

O conhecimento sobre a conformação de uma proteína é fundamental na determinação dos mecanismos e funções protéicas, podendo, por exemplo, ser utilizado na redução dos custos de desenvolvimento e teste de medicamentos (Andrew Pollack, "Drug Testers Turn to 'Virtual Patients' as Guinea Pigs", 10 de novembro de 1998, *New York Times*) . É evidente, portanto, por que tanto esforço é direcionado, atualmente, para o estudo dos problemas relacionados às proteínas.

Um grande volume de recursos tem sido aplicado no estudo das proteínas [5, 57]. A criação de bases de dados de estruturas protéicas como, por exemplo, o *Protein Data Bank* (PDB) [1], permite a coleta e armazenamento de todas as conformações de proteínas que foram identificadas nos últimos anos. Juntamente com as con-

formações tridimensionais, esta base de dados também fornece os dados brutos que foram utilizados para a obtenção da conformação, bem como alguns detalhes sobre os experimentos realizados.

No MDGP, no que diz respeito à complexidade, caso as distâncias entre todos os pares de átomos sejam previamente conhecidas, exite uma única estrutura tridimensional a ser determinada, obtida em tempo polinomial [14]. No entanto, apenas um subconjunto das distâncias pode ser obtido via experimentos de RMN. Neste caso, o problema passa a ser NP-difícil [52].

Existem, na literatura, várias abordagens para o MDGP. Por exemplo, o algoritmo *EMDED* de Crippen e Havel [12, 21], a estratégia de redução de grafo de Hendrickson [23, 24], o algoritmo DGSOL de Moré e Wu [43–46], o método de perturbação estocástica de Zou, Bird e Schnabel [60], o método de escalonamento multidimensional de Trosset [56], o algoritmo *Variable Neighborhood Search* (VNS) de Lavor, Liberti e Maculan [30, 36], o algoritmo *Geometric Build-Up* de Dong, Wu e Wu [59], a extensão do algoritmo *Geometric Build-Up* feita por Carvalho, Lavor e Protti [7], entre outros. Existem também alguns levantamentos sobre métodos para a resolução deste problema, que podem ser encontrados em [6, 12, 22, 31, 32, 39, 42].

Recentemente, foram propostas duas formulações combinatórias para o DGP. As discretizações do problema são possíveis quando algumas hipóteses particulares são satisfeitas. Em 2006, Lavor, Liberti e Maculan [29] propuseram a primeira formulação baseada na estrutura de conformações de proteínas, onde observou-se que é possível formular o MDGP, aplicado à cadeia principal de uma proteína, como um problema de busca em um espaço discreto. Essa nova formulação foi denotada por Problema Discreto de Geometria de Distâncias Moleculares (DMDGP, *Discretizable Molecular Distance Geometry Problem*). Mais recentemente, em 2011, Mucherino, Lavor e Liberti [48] propuseram uma outra formulação discreta para o DGP que se baseia em hipóteses mais fracas, que não estão relacionadas com conformações moleculares. Essa outra formulação foi denominada Problema Discreto de Geometria de Distâncias (DDGP, *Discretizable Distance Geometry Problem*). Em relação à complexidade, tanto o DMDGP quanto o DDGP são NP-difíceis, como provado, respectivamente, em [29] e [48]. Para resolver ambos os problemas combinatórios, foi empregado um algoritmo *Branch & Prune* (BP) [29, 37, 48], que é fortemente baseado na estrutura combinatória dos dois problemas.

O DDGP pode ser resolvido através do algoritmo Branch & Prune (BP) [29]. No entanto, para se aplicar o BP, as hipóteses do DDGP devem ser satisfeitas. Assim, encontrar uma ordem para os vértices V, do grafo associado, satisfazendo tais hipóteses, representa uma importante etapa de pré-processamento para a solução de DDGPs [48]. Dizemos que uma ordem para os vértices de V é válida para o DDGP se ela satisfaz as hipóteses do DDGP. Este problema é referido como o *Discretizing*

Vertex Order Problem (DVOP) [34]. Um algoritmo capaz de resolver o DVOP também foi proposto em [34].

Uma variação do DMDGP, o Problema Discreto de Geometria das Distâncias em Moléculas com Distâncias Intervalares (iDMDGP, *Interval Discretizable Molecular Distance Geometry Problem*), foi proposta em [33]. Ainda em [33], uma nova modificação do BP foi proposta para trabalhar com distâncias intervalares, o *Branch & Prune* Intervalar (iBP, *Interval Branch & Prune*). Nessa versão, consideram-se não somente distâncias exatas, mas também intervalares. Assim, tanto no BP clássico quanto no iBP, é importante a construção de uma ordem entre os átomos que satisfaçam as condições do DMDGP ou iDMDGP. Nesse sentido, em [33], foi proposta uma ordem para a cadeia principal da molécula de proteína que satisfaz as hipóteses do iDMDGP.

Nesta tese, é apresentada uma abordagem paralela para o DDGP. A estratégia usada para a paralelização do algoritmo é de simples entendimento, mas de difícil implementação. O algoritmo gerencia de forma eficiente o custo de tempo computacional, quando comparado com diferentes números de processos envolvidos na computação paralela. Para avaliar os resultados experimentais do algoritmo paralelo, foram geradas instâncias baseadas em informações provenientes do PDB. Apresentamos também uma nova ordem para a cadeia principal de uma molécula de proteína, tal que essa nova ordem satisfaz as condições do iDMDGP. Essa nova ordem, que envolve um número menor de átomos e explora algumas propriedades químicas da cadeia principal da proteína, é comparada com uma ordem proposta em [33]. Abordamos ainda, um novo algoritmo para o DVOP, ou seja um algoritmo capaz de ordenar os vértices de tal modo que atenda as hipóteses do DDGP. O novo algoritmo para o DVOP foi testado em instâncias do *Sensor Network Localization Problem* (SNLP) e comparado com um algoritmo proposto em [34].

O trabalho está organizado em sete capítulos. O conteúdo de cada capítulo é apresentado a seguir.

- Capítulo 2: introduz o Problema de Geometria das Distâncias em Moléculas (MDGP). Em seguida, o Problema Discreto de Geometria das Distâncias em Moléculas (DMDGP) é formalmente definido e, então, posteriormente, é apresentada a generalização desta discretização através do Problema Discreto de Geometria das Distâncias (DDGP);

- Capítulo 3: apresenta uma das contribuições desta tese: um algoritmo paralelo para o DDGP. Uma análise experimental, comparando o algoritmo paralelo proposto com diferentes números de processos envolvidos na computação paralela, é apresentada. No caso em que apenas um processo é considerado, o mesmo é equivalente ao algoritmo sequencial da literatura;

- Capítulo 4: mostra uma abordagem para o DMDGP, proposta em [33], considerando não apenas distâncias exatas, mas também distâncias intervalares. Esse abordagem é conhecida como o Problema Discreto de Geometria das Distâncias em Moléculas com distâncias intervalares (iDMDGP).

- Capítulo 5: aborda ordens artificias em moléculas de proteínas que satisfazem as condições do iDMDGP. Propomos uma nova ordem para os átomos da cadeia principal de uma molécula de proteína e comparamos com uma ordem proposta anteriormente na literatura.

- Capítulo 6: apresenta ordens válidas para o DDGP. O problema de encontrar ordens válidas para o DDGP é conhecido como DVOP. Assim, apresentamos um novo algoritmo para o DVOP que é capaz de encontrar uma ordem para o DVOP com menos tempo de CPU do que um algoritmo da literatura. Esse novo algoritmo é aplicado em instâncias do *Sensor Network Localization Problem* (SNLP).

- Capítulo 7: Finalmente, neste capítulo, apresentamos as conclusões e propomos alguns caminhos para trabalhos futuros.

Capítulo 2

Problema de Geometria de Distâncias Moleculares (MDGP)

2.1 Descrição do MDGP

O Problema de Geometria de Distâncias em Moleculares (MDGP, em inglês, *Molecular Distance Geometry Problem*) está associado à determinação da estrutura tridimensional de uma molécula [28]. Este problema pode ser formulado da seguinte maneira: encontre as posições $x_1, \ldots, x_n \in \mathbb{R}^3$ dos átomos da molécula, tais que

$$\| x_i - x_j \| = d_{i,j}, \qquad (i,j) \in S, \tag{2.1}$$

onde S é um subconjunto dos pares de átomos cujas distâncias $d_{i,j}$ são conhecidas a priori e $||.||$ é a norma Euclidiana.

A formulação (2.1) corresponde ao MDGP exato. Devido aos erros experimentais na análise de RMN, somente alguns limites inferiores e superiores das distâncias podem ser obtidos. Deste modo, o MDGP pode ser definido, de um modo mais geral, encontrando as posições $x_1, \ldots, x_n \in \mathbb{R}^3$ tais que

$$l_{i,j} \leq \| x_i - x_j \| \leq u_{i,j}, \qquad (i,j) \in S, \tag{2.2}$$

onde $l_{i,j}$ e $u_{i,j}$ são os limites inferiores e superiores nas restrições das distâncias, respectivamente.

O MDGP pode ser formulado como um problema de otimização contínua, onde a função objetivo é dada por

$$f(x_1, \ldots, x_n) = \sum_{(i,j) \in S} (||x_i - x_j||^2 - d_{i,j}^2)^2. \tag{2.3}$$

A grande dificuldade nessa formulação é que a quantidade de mínimos locais cresce

exponencialmente com o tamanho da molécula e o que se deseja é encontrar o mínimo global [23].

2.2 Problema Discreto de Geometria das Distâncias em Moléculas (DMDGP)

Como descrito anteriormente, o MDGP pode ser visto como um problema de otimização contínua. Entretanto, usando duas hipóteses adicionais comumente aplicáveis às estruturas protéicas, uma formulação discreta foi proposta em [29], introduzindo assim uma subclasse do MDGP, chamada de Problema Discreto de Geometria de Distâncias Moleculares (DMDGP, em inglês, *The Discretizable Molecular Distance Geometry Problem*). Também em [29], foi demostrado que o DMDGP é NP-difícil.

2.2.1 Formulação discreta

Considere uma molécula como sendo uma sequência de n átomos, onde os comprimentos de ligações covalentes, que corresponde à distância média entre os núcleos de dois átomos ligados na posição de maior estabilidade (menor energia), são denotadas por $d_{i-1,i}$, para $i = 2, \ldots, n$, os ângulos de ligações covalentes são denotados por $\theta_{i-2,i}$, para $i = 3, \ldots, n$, e os ângulos de torção denotados por $\omega_{i-3,i}$, para $i = 4, \ldots, n$. Os ângulos de torção são definidos pelos vetores normais dos planos definidos pelos átomos $i-3, i-2, i-1$ e $i-2, i-1, i$, respectivamente (Figura 2.1).

Para a formulação discreta do MDGP, são consideradas as seguintes hipóteses:

Hipótese A1: os comprimentos e os ângulos de ligações, bem como as distâncias entre átomos separados por 3 ligações consecutivas são conhecidos. Em termos de grafos, deve-se existir uma clique entre quaisquer 4 átomos (vértices) consecutivos, onde as arestas estão relacionadas ao fato de que as distâncias envolvidas são conhecidas.

Hipótese A2: Os ângulos de ligações não podem ser múltiplos de π. Ou seja, $d_{i-3,i-1} < d_{i-3,i-2} + d_{i-2,i-1}$.

A Hipótese A1 é aplicável à maioria das proteínas, pois os comprimentos de ligações e os ângulos de ligações são conhecidos a priori. Além disso, a RMN é capaz de obter distâncias entre átomos que estão próximos entre si, e grupos de quatro átomos consecutivos da cadeia principal de uma proteína são frequentemente próximos, com valores menores do que 6Å, que é a "precisão" da RMN [11, 54]. A

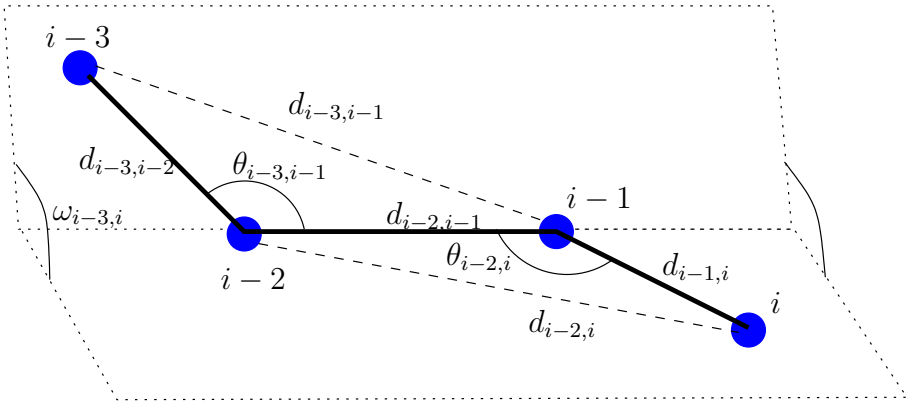

Figura 2.1: Definições de *comprimento de ligações*, *ângulos de ligações* e ângulos de torção (Figura retirada de [29]).

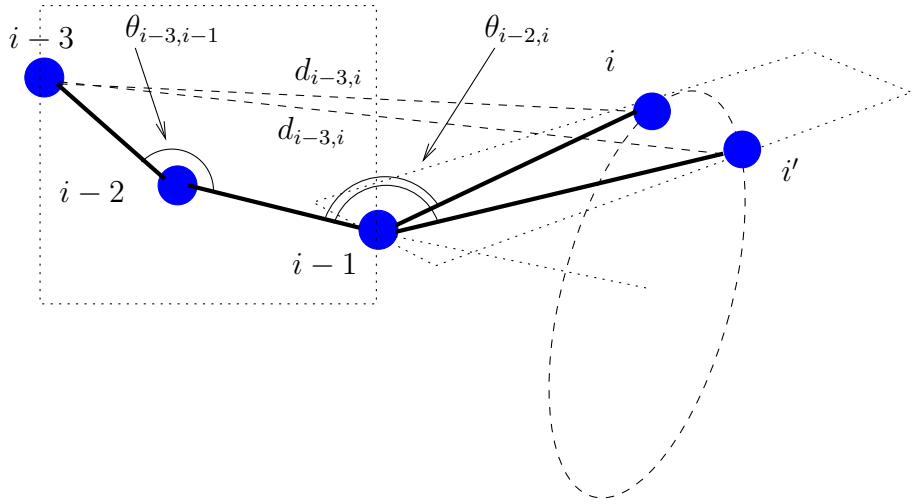

Figura 2.2: No DMDGP, o átomo i pode estar somente em duas posições (i e i') para ser "viável" com a distância $d_{i-3,i}$ (Figura retirada de [29]).

Hipótese A2 é igualmente aplicável às proteínas, dado que não se conhece proteína com ângulos de ligações covalentes com valor exato de π.

A intuição da formulação discreta é que o i-ésimo átomo reside na intersecção de três esferas centradas nos átomos $i-3, i-2, i-1$, de raios $d_{i-3,i}, d_{i-2,i}, d_{i-1,i}$, respectivamente. Pela Hipótese A2 e pelo fato de dois átomos não poderem nunca assumir a mesma posição no espaço, a intersecção das três esferas define, no máximo, dois pontos (indexados por i e i' na Figura 2.2). Isto permite expressar a posição do i-ésimo átomo em termos dos últimos três, dando-nos 2^{n-3} possíveis moléculas.

Dados todos os comprimentos de ligações $d_{1,2}, \ldots, d_{n-1,n}$, ângulos de ligações $\theta_{13}, \ldots, \theta_{n-2,n}$, e ângulos de torção $\omega_{1,4}, \ldots, \omega_{n-3,n}$ de uma molécula com n átomos, as coordenadas cartesianas $x_i = (x_{i_1}, x_{i_2}, x_{i_3})$, para cada átomo i na molécula, podem

ser obtidas utilizando a seguinte fórmula [51]:

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

onde

$$B_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad B_2 = \begin{bmatrix} -1 & 0 & 0 & -d_{1,2} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{2.4}$$

$$B_3 = \begin{bmatrix} -\cos\theta_{1,3} & -\sin\theta_{1,3} & 0 & -d_{2,3}\cos\theta_{1,3} \\ \sin\theta_{1,3} & -\cos\theta_{1,3} & 0 & d_{2,3}\sin\theta_{1,3} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

e

$$B_i = \begin{bmatrix} -\cos\theta_{i-2,i} & -\sin\theta_{i-2,i} & 0 & -d_{i-1,i}\cos\theta_{i-2,i} \\ \sin\theta_{i-2,i}\cos\omega_{i-3,i} & -\cos\theta_{i-2,i}\cos\omega_{i-3,i} & -\sin\omega_{i-3,i} & d_{i-1,i}\sin\theta_{i-2,i}\cos\omega_{i-3,i} \\ \sin\theta_{i-2,i}\sin\omega_{i-3,i} & -\cos\theta_{i-2,i}\sin\omega_{i-3,i} & \cos\omega_{i-3,i} & d_{i-1,i}\sin\theta_{i-2,i}\sin\omega_{i-3,i} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{2.5}$$

para $i = 4, ..., n$.

Para cada quatro átomos consecutivos $x_{i-3}, x_{i-2}, x_{i-1}, x_i$, o cosseno do ângulo de torção $\omega_{i-3,i}$ para $i = 4, ..., n$, pode ser determinado por:

$$\cos\omega_{i-3,i} = \frac{d_{i-3,i-2}^2 + d_{i-2,i}^2 - 2d_{i-3,i-2}d_{i-2,i}\cos\theta_{i-2,i}\cos\theta_{i-1,i+1} - d_{i-3,i}^2}{2d_{i-3,i-2}d_{i-2,i}\text{sen}\theta_{i-2,i}\text{sen}\theta_{i-1,i+1}}, \tag{2.6}$$

que é apenas um rearranjo da lei dos cossenos para os ângulos de torção [29].

Usando os comprimentos de ligações $d_{1,2}, d_{2,3}$ e o ângulo de ligação $\theta_{1,3}$, podemos

calcular as matrizes B_2 e B_3, definidas em (2.4), e obter:

$$\begin{aligned} x_1 &= \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \\ x_2 &= \begin{pmatrix} -d_{1,2} \\ 0 \\ 0 \end{pmatrix}, \\ x_3 &= \begin{pmatrix} -d_{1,2} + d_{2,3}\cos\theta_{1,3} \\ d_{2,3}\sin\theta_{1,3} \\ 0 \end{pmatrix}, \end{aligned}$$

fazendo com que os três primeiros átomos da molécula sejam fixados, pela Hipótese A1.

Uma vez que a distância $d_{1,4}$ é conhecida, novamente pela Hipótese A1, o valor de $\cos\omega_{1,4}$ pode ser obtido. Assim, o seno do ângulo de torção $\omega_{1,4}$ pode ter apenas dois valores possíveis: $\sin\omega_{1,4} = \pm\sqrt{1-\cos^2\omega_{1,4}}$. Deste modo, por (2.5), obtemos apenas duas posições possíveis (x_4, x_4') para o quarto átomo da molécula:

$$\begin{aligned} x_4 &= \begin{bmatrix} -d_{1,2} + d_{2,3}\cos\theta_{1,3} - d_{3,4}\cos\theta_{1,3}\cos\theta_{2,4} + d_{3,4}\sin\theta_{1,3}\sin\theta_{2,4}\cos\omega_{1,4} \\ d_{2,3}\sin\theta_{1,3} - d_{3,4}\sin\theta_{1,3}\cos\theta_{2,4} - d_{3,4}\cos\theta_{1,3}\sin\theta_{2,4}\cos\omega_{1,4} \\ d_{3,4}\sin\theta_{2,4}\left(\sqrt{1-\cos^2\omega_{1,4}}\right) \end{bmatrix}, \\ x_4' &= \begin{bmatrix} -d_{1,2} + d_{2,3}\cos\theta_{1,3} - d_{3,4}\cos\theta_{1,3}\cos\theta_{2,4} + d_{3,4}\sin\theta_{1,3}\sin\theta_{2,4}\cos\omega_{1,4} \\ d_{2,3}\sin\theta_{1,3} - d_{3,4}\sin\theta_{1,3}\cos\theta_{2,4} - d_{3,4}\cos\theta_{1,3}\sin\theta_{2,4}\cos\omega_{1,4} \\ d_{3,4}\sin\theta_{2,4}\left(-\sqrt{1-\cos^2\omega_{1,4}}\right) \end{bmatrix}. \end{aligned}$$

Para o quinto átomo, obtemos quatro possíveis posições, uma para cada combinação de $\pm\sqrt{1-\cos^2\omega_{1,4}}$ e $\pm\sqrt{1-\cos^2\omega_{2,5}}$. Por indução, podemos observar que para o i-ésimo átomo, existem 2^{n-3} posições possíveis. Desta forma, para representarmos uma molécula como uma sequência linear de n átomos, temos 2^{n-3} possíveis sequências de ângulos de torção $\omega_{1,4}, \ldots, \omega_{n-3,n}$, cada uma definindo uma diferente estrutura tridimensional. Utilizando as matrizes B_i (2.5), essa sequência de ângulos de torção pode ser convertida em uma outra sequência de coordenadas cartesianas $x = (x_1, \ldots, x_n) \in \mathbb{R}^3$.

Propriedade do posicionamento único

Como ilustrado na Figura 2.2, uma vez que os átomos $i-3$, $i-2$ e $i-1$ estão fixos, existem sempre duas possíveis posições para o átomo i. Contudo, foi observado que existem alguns casos particulares, onde há somente uma posição possível para se

colocar este átomo.

Utilizando como exemplo as duas posições possíveis (x_4 e x'_4) para o quarto átomo, apresentadas anteriormente, pode-se verificar que a única diferença entre elas está na coordenada z. Entretanto, se a igualdade $\sqrt{1-\cos^2\omega_{1,4}} = -\sqrt{1-\cos^2\omega_{1,4}}$ for satisfeita, as posições (x_4 e x'_4) são idênticas, ou seja existe somente uma posição para o quarto átomo. Esta igualdade é verdadeira quando $\cos^2\omega_{1,4} = 1$, e isso ocorre quando $\omega_{1,4}$ é múltiplo de π.

De forma genérica, pode-se definir essa propriedade como: para todo átomo i, se a igualdade $\sqrt{1-\cos^2\omega_{i-3,i}} = -\sqrt{1-\cos^2\omega_{i-3,i}}$ for verdadeira, existe somente uma posição viável para i, uma vez que $x_i = x'_i$.

Soluções simétricas

Em [29], onde o DMDGP é definido, foi demostrado que para qualquer solução S do problema, existe uma solução S' simétrica a S. Esta simetria ocorre em relação ao plano definido pelos três primeiros átomos que são fixados, sendo que qualquer solução de um "lado" deste plano dá origem a uma solução simétrica do outro "lado". A prova matemática deste teorema pode ser vista em [29]. No entanto, neste trabalho optou-se por mostrar somente um exemplo visual (Figura 2.3). Na Figura 2.3, são mostradas duas soluções simétricas para DMDGP. Nas duas soluções, os três primeiros átomos estão nas posições 1, 2 e 3, respectivamente, já o quarto, quinto e sexto átomos se encontram em posições distintas em cada uma das soluções. Para uma delas, estes átomos estão nas posições 4, 5 e 6, para a outra, eles estão em $4'$, $5'$ e $6'$ respectivamente. Em ambas as soluções, a distância entre quaisquer par de átomos é a mesma. Desta forma, se uma delas é válida, a outra também é. Com isso, ao se encontrar uma solução para o problema, pode-se gerar uma solução simétrica a esta.

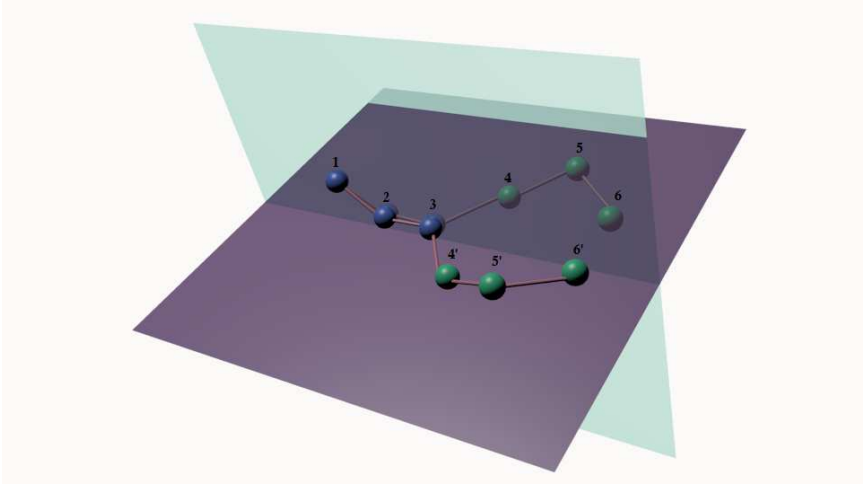

Figura 2.3: Simetria das soluções do DMDGP.

2.3 Problema Discreto de Geometria das Distâncias (DDGP)

Recentemente, em [48], foi proposta uma generalização do DMDGP para dimensões superiores a três: o Problema Discreto de Geometria de Distâncias ou, conforme em inglês, *The Discretizable Distance Geometry Problem* (DDGP). No caso tridimensional (DDGP3), a principal diferença do DMDGP está na hipótese de discretização. Em vez dos três predecessores imediatos de v, quaisquer três vértices anteriores a v podem ser considerados. Disso resulta que o DDGP3 depende de hipóteses mais fracas que as do DMDGP. Em particular, a hipótese de discretização do DDGP não reflete qualquer recurso de moléculas ou proteínas e, portanto, o DDGP pode ser considerado como um problema mais genérico que pode ser empregado em outras aplicações. Ainda em [48], foi demostrado que qualquer instância DMDGP é também uma instância do DDGP3. Entretanto, a recíproca não é verdadeira, isto é, existem instâncias DDGP3 que não são instâncias do DMDGP para um dada ordem dos vértices. Como o DDGP3 contém o DMDGP, e o DMDGP é NP-difícil [29, 48], o DDGP3 também é NP-difícil. A seguir, é apresentada uma definição formal do problema.

2.3.1 Formulação discreta

Esta seção mostrará uma formulação discreta para o DGP, cuja definição pode ser vista a seguir:

Definição 2.1 *The Discretizable Distance Geometry Problem (DDGP) [48].*
Considere $G = (V, E, d)$ um grafo ponderado não-direcionado associado a uma instância do DGP. Suponha que existe uma relação de ordem parcial dos vértices em V. O DDGP na dimensão k consiste em todas instâncias do DGP que satisfazem as seguintes hipóteses:

Hipótese B1: *Existe um sub-conjunto V_1 de V tal que:*

- $|V_1| = K + 1$;
- *a relação de ordem em V_1 é total;*
- *V_1 é uma clique;*
- $\forall v_0 \in V_1 \quad \forall v \in V \setminus V_1,\ v_0 < v$.

Hipótese B2: $\forall v \in V \setminus V_1,\ \exists u^1, u^2, \ldots, u^K \in V$ *tal que:*

- $u^1 < v, u^2 < v, \ldots, u^K < v$;
- $\{(u^1, v), (u^2, v), \ldots, (u^K, v)\} \in E$;

- *O determinante de Cayley-Menger* [1] *da matriz de distância relacionada a* $\{u^1, u^2, \ldots, u^K\}$ *é diferente de zero.*

Pela definição 2.1 do DDGP, apenas uma relação de ordem parcial é necessária sob os vértices de G. Entretanto, no caso do DDGP3 (ou seja, com $K = 3$), observa-se que o conjunto de todas as ordens parciais pode ser estendido para uma ordem total. Pela hipótese B2, para cada vértice v, deve existir, pelo menos, 3 vértices u^1, u^2 e u^3 que precede v e tal que as distâncias $d(u^1, v)$, $d(u^2, v)$ e $d(u^3, v)$ sejam conhecidas. Esta hipótese é mais fraca que a hipótese análoga do DMDGP (hipótese A1 que também exige que os quatro vértices u^1, u^2, u^3 e v sejam consecutivos).

Considere três esferas, centradas em x_{u^1}, x_{u^2}, x_{u^3}, e com raios $d(x_k, x_{u^1})$, $d(x_k, x_{u^2})$, $d(x_k, x_{u^3})$, respectivamente. A intersecção dessas três esferas fornece um conjunto de posições possíveis para x_k, ou seja, posições que respeitam as três distâncias entre k e u^1, u^2, u^3. A intuição da formulação discreta é que a intersecção entre as esferas pode ser um círculo, dois pontos ou apenas um ponto. Entretanto, o círculo é obtido se os três vértices u^1, u^2 e u^3 estiverem alinhados, o que não é permitido pela desigualdade triangular estrita (hipótese B2). Assim, em todos os casos, há, no máximo, duas posições para o vértice k. Isto permite expressar a posição do k-ésimo vértice em termos de outros tês anteriores quaisquer, dando-nos 2^k possíveis posições. Se considerarmos que os três primeiros vértices sejam fixos, temos então, 2^{k-3} possíveis posições.

Resolver o problema de encontrar a intersecção de três esferas consiste em determinar as duas posições para um dado vértice k, sendo equivalente ao problema de encontrar as duas soluções do seguinte sistema de equações quadráticas:

$$\begin{cases} \|x_k - x_{u^1}\| = d(x_k, x_{u^1}) \\ \|x_k - x_{u^2}\| = d(x_k, x_{u^2}) \\ \|x_k - x_{u^3}\| = d(x_k, x_{u^3}) \end{cases} \tag{2.7}$$

Métodos para encontrar soluções para o sistema (2.7) podem ser encontrados, por exemplo, em [9]. Pode-se destacar que, seja qual for o método utilizado, é muito importante que as soluções encontradas sejam muito precisas. Na verdade, elas representam as posições possíveis para os vértices dos grafos que satisfazem alguns testes de viabilidade antes de serem inseridos na árvore binária. Portanto, se as soluções encontradas para (2.7) não forem precisas o suficiente, então os testes de poda podem rejeitar todas elas e não serem encontradas soluções.

[1] Em geral, para determinar uma estrutura no espaço euclidiano de n dimensões certas relações (restrições) entre as distâncias devem existir. Os determinantes Cayley-Menger são, então, usados para caracterizar os espaços euclidianos em termos de distâncias entre pontos. [55].

2.4 Algoritmo Branch & Prune

Conforme descrito anteriormente, ambas as hipóteses do DMDGP e do DDGP permitem a discretização do DGP. Considere que as posições para os vértices $i \in \{1, \ldots, k-1\}$ de uma solução para o problema já estejam determinadas e que a posição para o k-ésimo vértice é buscada. Pelas hipóteses do DMDGP e do DDGP, exitem três vértices u^1, u^2 e u^3 tal que as distâncias entre k e u^1, u^2, u^3 são conhecidas. No caso do DMDGP, os três vértices u^1, u^2 e u^3 são aqueles que imediatamente precedem k. No caso do DDGP, cada vértice u^1, u^2 e u^3 pode ser qualquer vértice com índice menor que k. Em ambos os casos, a distância entre k e os três outros vértices, cuja posições são conhecidas, pode ser usado para computar as possíveis posições para k.

Intuitivamente, baseando-se na estrutura combinatória do DMDGP e do DDGP, onde em cada iteração o k-ésimo vértice pode ser posicionado no máximo em duas possíveis posições, chamemos de $x_k^{(0)}$ ou $x_k^{(1)}$. Mais especificamente, a estrutura dos problemas pode ser representada em uma árvore binária, como exemplificado na Figura 2.4 com 6 vértices. Neste exemplo, considera-se que os vértices k, $k+1$ e $k+2$ sejam fixos e que os vértices $k+3$, $k+4$ e $k+5$ possam ser colocados em duas, quatro e oito possíveis posições, respectivamente.

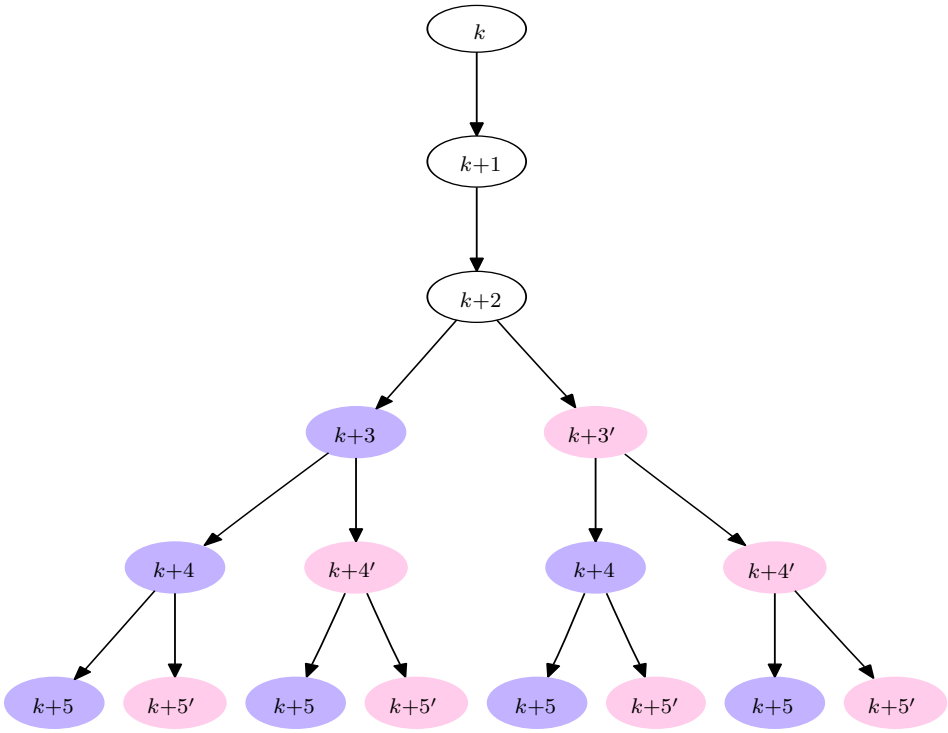

Figura 2.4: Representação em árvore binária

O algoritmo Branch & Prune (BP), proposto em [37], explora de forma eficiente esta árvore binária. A árvore não é construída a priori, mas sim durante o processo

de busca. A cada passo do algoritmo, duas novas posições são computadas para o k-ésimo vértice. Elas serão adicionadas à árvore somente se satisfazerem alguns testes de viabilidade. Na verdade, as duas posições são computadas de forma que satisfaçam as distâncias conhecidas entre k e os três vértices u^1, u^2, u^3. No entanto, pode haver outras distâncias disponíveis que podem ser utilizadas para verificar a viabilidade das posições encontradas.

O teste de poda mais simples e natural é aquele em que as distâncias conhecidas e as distâncias obtidas a partir das posições calculadas para o vértice k são comparadas. Para isso, verifica-se

$$(||x_k - x_j||^2 - d_{kj}^2)^2 < \varepsilon, \tag{2.8}$$

onde $\varepsilon > 0$ é uma tolerância dada. Diante disso, as seguintes situações podem ocorrer: a posição verificada é viável e satisfaz a desigualdade, ou inviável, não satisfazendo a desigualdade. Neste último caso, a posição não é adicionada à árvore e todas as posições ao longo do mesmo ramo da árvore não são consideradas, porque elas não podem ser parte de uma solução viável. Assim, o algoritmo deve percorrer, de alguma forma, essa estrutura de árvore e realizar podas nos nós onde a posição aferida para o vértice é incorreta. Durante o percurso, quando uma folha é alcançada e esta tem uma posição que respeita as restrições de distância, uma solução para o problema é encontrada.

Esta fase de poda no algoritmo BP permite reduzir a árvore binária muito rapidamente, de modo que uma pesquisa exaustiva sobre os ramos restantes não é muito cara. O Algoritmo 1 representa o pseudo-código do algoritmo BP.

O algoritmo BP, com pequenos ajustes, pode resolver, de forma eficiente, instâncias do DMDGP e DDGP relacionadas com conformações de proteína, como descrito respectivamente em [29, 37, 48]. No caso de instâncias DMDGP, a árvore binária é construída através do calculo do ângulos de torção. No caso de instâncias do DDGP, a árvore binária é construída através da solução de sistemas de equações quadráticas.

Em [37, 48], são encontrados resultados numéricos obtidos com o algoritmo BP e com dados artificiais propostos por Moré Wu [45] e Lavor [28] e dados reais obtidos no PDB. Uma versão não recursiva, que gerencia de forma eficiente o tempo de CPU e o uso de memória do algoritmo BP foi proposto em [17].

Algoritmo 1 Branch & Prune

1: BP(k, n, d)
2: calcule a primeira posição para o k-ésimo átomo: $x_k^{(0)}$;
3: verifique a viabilidade da posição $x_k^{(0)}$:
4: **if** $((\|x_k^{(0)} - x_j\|^2 - d_{kj}^2)^2 < \epsilon,\ \forall j < k)$ **then**
5: **if** $k = n$ **then**
6: uma solução foi encontrada
7: $nsol \leftarrow nsol + 1$;
8: $sol(nsol, *)$ lista das soluções encontradas
9: **else**
10: BP($k + 1$, n, d)
11: **end if**
12: **else**
13: a posição $x_k^{(0)}$ é podada.
14: **end if**
15: calcule a segunda posição para o i-ésimo átomo: $x_i^{(1)}$;
16: verifique a viabilidade da posição $x_i^{(1)}$:
17: **if** $((\|x_k^{(1)} - x_j\|^2 - d_{kj}^2)^2 < \epsilon,\ \forall j < k)$ **then**
18: **if** $i = n$ **then**
19: uma solução foi encontrada
20: $nsol \leftarrow nsol + 1$;
21: $sol(nsol, *)$ lista das soluções encontradas
22: **else**
23: BP($k + 1$, n, d)
24: **end if**
25: **else**
26: a posição $x_k^{(1)}$ é podada.
27: **end if**
28: **return** sol;

Capítulo 3

Branch & Prune Paralelo

Uma versão paralela do BP para o DMDGP foi proposto em [47]. A idéia principal é dividir uma determinada instância em p sub-instâncias a serem atribuídas em p processos diferentes envolvidos na computação paralela. Em cada processo, o algoritmo BP sequencial é invocado para encontrar todas as soluções de cada sub-instância, e cada solução é armazenada na memória em formato binário (escolha do ramo esquerda/direito em cada nível da árvore binária). Cada processo envia suas soluções locais para outros processos de uma forma hierárquica como mostrado na Figura 3.1, e, sequencialmente, todos os processos trabalham na construção do conjunto final de soluções (informações sobre as distâncias entre os vértices previamente atribuídas a diferentes processos são utilizadas para remover soluções inviáveis). Este último passo aponta o ponto fraco deste algoritmo paralelo. As coordenadas calculadas por cada processo são representadas em sistemas independentes de coordenadas e, portanto, não podem ser reutilizados nesta etapa final, onde todas as coordenadas são recalculadas, em vez de um sistema de coordenadas comum explorando as informações sobre as soluções locais recebidas.

3.1 BP Paralelo para o PDGD

Nesta seção, apresentamos o algoritmo que propomos neste trabalho, ou seja, uma versão paralela do algoritmo BP para o DDGP, em que as coordenadas finais contidas nas soluções são geradas diretamente pelos diferentes processos envolvidos no cálculo. Isto é possível porque o mesmo sistema de coordenadas é utilizado por todos os processos durante as chamadas para os BPs sequenciais, e alguns vértices são atribuídos a todos os processos, sendo que, para que isso ocorra, a ordenação dos vértices precisa ser modificada (isso seria praticamente impossível para o DMDGP). Como consequência, nesta versão paralela do BP para o DDGP, a construção do conjunto final de soluções é menos caro, porque todas as coordenadas necessárias são recebidas de outros processos e não precisamos recalculá-las. A tarefa é redu-

zida apenas à identificação de soluções inviáveis (através de informações sobre as distâncias entre os vértices previamente atribuídas a diferentes processos).

De forma análoga à versão paralela do BP para o DMDGP, a idéia principal da versão paralela do algoritmo BP para DDGP é dividir uma instância DDGP em sub-instâncias tantas quanto o número de processos envolvidos na computação paralela, e para resolver cada sub-instância, são usadas chamadas locais para BPs sequenciais. Como no DDGP não há hipótese de consecutividade nos vértices que são considerados para a definição das três esferas a serem interceptadas, subconjuntos de vértices não consecutivos podem ser identificados e, em seguida, cada um deles pode ser atribuído a um único processo. Com a finalidade de fazer cada BP local trabalhar em um sistema de coordenadas comum, os três primeiros vértices na ordem que estão associados a cada subconjunto devem ser comuns a todos as sub-instâncias. Para todos os vértices associados a uma sub-instância, deve ser válida uma ordem que satisfaça as hipóteses de discretização do DDGP.

Seja $G = (V, E, d)$ um grafo não-direcionado ponderado representando uma instância da DDGP.

Definição 3.1 *Instância p-paralelizável*
O Grafo G representa uma instância p-paralelizável do DDGP, se, e somente se, existem p subconjuntos de vértices $\{V_1, V_2, \ldots, V_p\}$ *cobrindo* V *em um subconjunto* V_0 *com cardinalidade* 3 *tais que*

- $V_1 \cap V_2 \cap \cdots \cap V_p = V_0$,
- $\forall v_0 \in V_0,\ \forall i \in \{1, 2, \ldots, P\},\ \forall v \in V_i,\ v_0 < v$,
- *existe uma ordem que permite a discretização em cada subconjunto* V_i.

A definição de uma instância p-paralelizável garante que a instância original pode ser dividida em p sub-instâncias que pertencem à classe DDGP (de modo que o BP pode ser invocado para resolvê-las) e que, em todas as p ordens associadas às sub-instâncias, os três primeiros vértices são aqueles em V_0 (de modo que todas as soluções são construídas no mesmo sistema de coordenadas). Nessas hipóteses, o conjunto de soluções locais obtido pelos BPs locais pode ser transmitido para outros processos, e o conjunto final de soluções pode ser obtido combinando soluções locais. Durante esta etapa, é importante verificar se as distâncias entre os vértices conhecidos previamente atribuídas a diferentes processos são satisfeitas. Caso contrário, a solução correspondente precisa ser removida do conjunto final.

Após a execução dos algoritmos BPs locais em paralelo, os conjuntos de soluções parciais locais precisam ser coletados e distribuídos aos processos. Para este objetivo, consideramos o esquema "cascata" clássica para as comunicações necessárias entre os

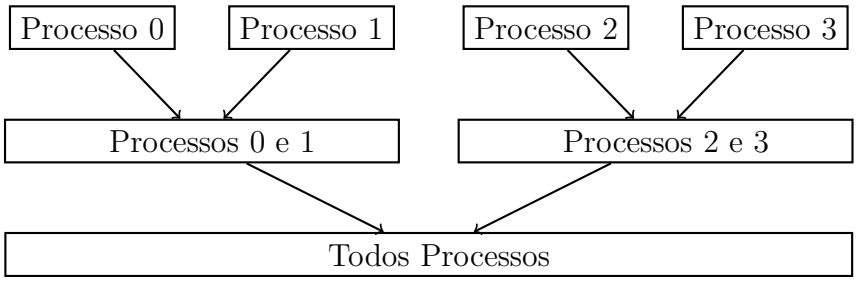

Figura 3.1: Esquema clássico de comunicação: "cascata"

processos (ver Figura 3.1). Esse esquema de comunicação é muito eficiente quando o número de processadores considerados é uma potência de 2, o que permite a divulgação de informações de um processo para todos os outros com um número de fases de comunicação que é igual a $\log_2(p)$.

Durante cada fase de comunicação, os pares de processos estabelecem uma comunicação no intuito de trocar informações locais encontradas pelos BPs sequenciais, ou seja, as coordenadas de suas soluções. Após cada fase, portanto, diferentemente do algoritmo paralelo para o DMDGP apresentado em [47], começamos a combinar soluções locais, que podem gerar soluções inviáveis antes de continuar a troca de informação local. Por exemplo, após a primeira fase de comunicação (ver Figura 3.1), as soluções encontradas pelos processos 0 e 1 podem ser combinadas, bem como as soluções encontradas pelos processos 2 e 3. Isso produziria dois novos conjuntos de soluções locais (soluções inviáveis podem ser descobertas e descartadas), que poderiam ser trocados na próxima fase de comunicação. Por esta razão, o esquema cascata é muito mais eficiente para este novo algoritmo paralelo.

O Algoritmo 2 é um pseudo-código desta versão paralela do BP para o DDGP. A lista de parâmetros de entrada contém os parâmetros necessários para o BP (ver Algoritmo 1), bem como o parâmetro p, que indica o número de processos envolvidos na computação paralela. Uma vez que a instância na entrada é dividida em p sub-instâncias, as p execuções paralelas da versão sequencial do BP são chamadas a fim de resolver suas sub-instâncias locais. Então, a troca de informações locais é realizada através da aplicação do esquema de cascata (ver Figura 3.1), e, logo após cada fase de comunicação, as soluções locais e as soluções recebidas são combinadas e a viabilidade das soluções resultantes são verificadas. O novo conjunto de soluções obtidas, em um dado processo, é então considerado durante a fase de comunicação sucessivas. No final, após todas as fases de comunicação, um conjunto completo de soluções é obtido.

Como descrito na seção 2.4, o algoritmo BP explora a estrutura de árvore binária para resolver o DDGP. Neste sentido, o algoritmo paralelo BP para o DDGP explora parte desta árvore binária em paralelo. A Figura 3.2 apresenta uma representação

Algoritmo 2 Branch & Prune Paralelo

```
1: parBP( i, n, d, p )
2: divide a instância em p sub-instâncias:
3: calcule i^(k), n^(k), d^(k) (k = 0, ..., p − 1);
4: call local BP( i^(k), n^(k), d^(k) ); (ver Algoritmo 1)
5: for k = 0, ..., log2(p) do
6:    execute fase k do esquema cascata
7:    for cada solução local x do
8:       for cada solução recebida y do
9:          combine as soluções x e y e crie z
10:         if ( z não é viável ) then
11:            descarte z
12:         end if
13:         descarte y
14:      end for
15:      descarte x
16:   end for
17: end for
```

desta árvore, no caso em que quatro processos são considerados. Para facilitar a representação, supomos que todas as chamadas locais do BP fornecem 2 soluções e que não houveram remoções de soluções inviáveis nas fases de comunicação, embora, em geral, o número de soluções encontradas por cada processo pode ser diferente e que soluções podem ser removidas quando as soluções de diferentes processos são combinadas.

3.2 Resultados Computacionais

Nesta seção, serão apresentados os experimentos computacionais para analisar o algoritmo paralelo proposto nesta tese. Primeiramente, são apresentadas as instâncias utilizadas para a realização dos testes e as métricas utilizadas para mensurar a qualidade das soluções. Em seguida, compara-se o desempenho do BP paralelo para o DDGP com o desempenho do algoritmo sequencial, bem como com diferentes números de processos. Para a realização dos experimentos, foram utilizadas instâncias reais extraídas do *Protein Data Bank* - PDB [1].

Os experimentos computacionais foram realizados em um *cluster* com 23 computadores, sendo 14 computadores Intel(R) Pentium(TM)D CPU 2.80Ghz com 4Gb e 9 computadores Intel(R) Core(TM)2 Quad CPU Q9550 @2.83Ghz com 4Gb, todos usando o sistema operacional Linux, versão 2.6.24-30.

O Branch & Prune paralelo proposto foi implementado em linguagem de programação C. Usou-se a biblioteca *Message Passing Interface* (MPI) [20, 41], versão MPICH2 1.0.5p4, e o compilador GNU C versão 4.2.4.

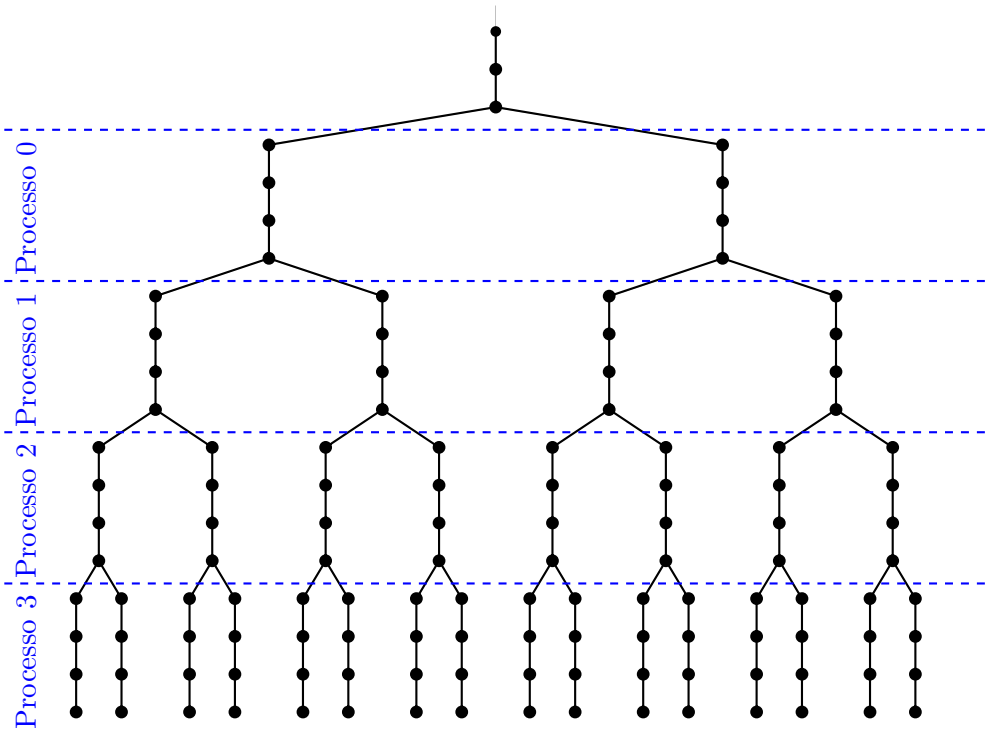

Figura 3.2: Representação em árvore do conjunto final de soluções obtidas pela combinação das soluções locais encontradas nos 4 processos.

3.2.1 Instâncias

As instâncias reais utilizadas foram geradas através de proteínas obtidas do PDB e podem ser livremente acessadas através do *site* `http://www.rcsb.org/pdb/`.

O PDB é um repositório, onde estruturas tridimensionais de proteínas são disponibilizadas. Todas as distâncias entre os átomos pertencentes a molécula podem ser obtidas, a partir dessas estruturas, uma vez que a posição de cada um deles é disponibilizada. Em cada uma das estruturas de proteínas, considerou-se o subconjunto de átomos de hidrogênio, onde todas as distâncias relativas entre pares de hidrogênios são calculadas, considerando apenas as distâncias com valores de até 6 Ångström (Å). A escolha por esse comprimento nas distâncias foi feita porque, segundo [54], para simular os dados obtidos a partir de experimentos da RMN, o comprimento de distância considerado deve ser, no máximo, igual a 6Å. Algumas distâncias maiores que 6Å podem ter sido incluídas para pares de átomos relativos ao subconjunto V_0 (ver Definição 3.1).

Foram utilizadas 10 diferentes instâncias consideradas grandes, com número de átomos variando entre 1000 e 2259. Na Tabela 3.1, as instâncias são descritas em detalhes: a primeira coluna apresenta uma numeração para a instância, a segunda coluna apresenta o nome da proteína (código PDB), a terceira coluna indica o número n de átomos e a quarta coluna indica o número de distâncias conhecidas.

Instâncias PDB			
Instância	Proteína	n	$\|E\|$
1	2KTU	1008	16681
2	1Q80	1054	18010
3	2K7N	1184	20330
4	1LA3	1193	19782
5	1E3D	1496	26308
6	1BST	1531	27981
7	1D8V	1563	26509
8	2K4T	1682	25557
9	2ROQ	2007	32975
10	1EZO	2259	35553

Tabela 3.1: Instâncias

3.2.2 Métricas de Qualidade das Soluções

Para que a qualidade das soluções possa ser mensurada, os algoritmos implementados utilizam duas conhecidas métricas da literatura: LDE e RMSD.

3.2.3 *Largest Distance Error* (LDE)

O LDE é empregado como uma medida de precisão da solução. Basicamente, ele compara as distâncias entre átomos da estrutura determinada com as distâncias conhecidas previamente. O LDE é definido como:

$$\text{LDE} = \frac{1}{|E|} \sum_{(i,j)\in E} \frac{|\,||x_i - x_j|| - d_{ij}|}{d_{ij}}, \tag{3.1}$$

onde E é o conjunto de todas as distâncias conhecidas. Quanto menor o LDE, melhor é a qualidade da solução.

3.2.4 *Root-Mean-Square Deviation* (RMSD)

Para comparar as estruturas encontras pelos algoritmos com as estruturas existentes no PDB, foi utilizado o cálculo do RMSD que, de maneira geral, mede o grau de semelhança entre duas estruturas [16]. O RMSD de duas estruturas X e Y pode ser definido da seguinte forma:

$$RMSD(X,Y) \;=\; \min_{Q} ||X - YQ||/\sqrt{n}, \tag{3.2}$$

onde Q é a matriz utilizada para rotacionar Y de modo que fique o mais semelhante possível de X.

O valor RMSD é expresso em unidades de comprimento. A unidade mais comumente utilizada em biologia estrutural é Ångström (Å).

3.2.5 Testes com o BP paralelo

Na Tabela 3.2, são apresentados os resultados dos experimentos computacionais para a versão paralela do BP para o DDGP. Consideramos moléculas de proteínas conforme descrito na Tabela 3.1. Para cada instância, p indica a quantidade de processadores envolvidos na computação paralela (p é uma potência de dois, com $1 \leq p \leq 128$, onde $p = 1$ indica que o BP sequencial foi utilizado), RMSD e LDE mostram o melhor valor encontrado para estas métricas, respectivamente, CPU time apresenta o tempo computacional, em segundos, utilizado pelo método para encontrar todas as soluções e #Sol é a quantidade de soluções encontradas pelo método.

Instância		parBP			
Número	p	LDE	RMSD	CPU time	#Sol
1	1	9.04e-11	1.37e-07	8.62	2
	2	3.35e-10	0.00e+00	2.93	2
	4	1.64e-10	7.11e-15	2.57	2
	8	1.11e-10	1.35e-14	1.30	2
	16	7.21e-08	1.95e-14	1.37	2
	32	4.19e-11	1.37e-07	0.19	2
	64	1.52e-08	1.37e-07	0.16	2
	128	5.21e-09	9.54e-15	0.30	2
2	1	9.45e-11	6.65e-15	14.01	2
	2	2.11e-06	3.23e-12	8.77	2
	4	2.23e-10	2.59e-12	2.53	2
	8	3.21e-09	1.53e-14	1.67	2
	16	4.87e-11	5.38e-15	0.25	2
	32	1.07e-10	7.61e-15	0.30	2
	64	3.98e-11	4.84e-15	0.19	2
	128	1.08e-10	1.40e-07	0.14	2
3	1	2.16e-10	1.16e-14	18.51	2
	2	3.05e-10	1.16e-14	4.58	2
	4	5.14e-10	1.40e-14	2.06	2
	8	1.33e-08	3.76e-14	1.15	2
	16	9.46e-09	1.65e-07	0.51	2
	32	2.74e-09	8.59e-15	0.32	2
	64	9.41e-11	1.65e-07	0.68	2

Instância		parBP			
Número	p	LDE	RMSD	CPU time	#Sol
	128	4.96e-11	1.57e-15	0.20	2
4	1	4.58e-10	3.15e-14	26.45	2
	2	2.83e-10	1.57e-07	10.25	2
	4	2.23e-10	1.57e-07	1.44	2
	8	2.56e-10	9.50e-15	1.47	2
	16	6.45e-10	1.57e-07	0.91	2
	32	9.25e-10	1.57e-07	0.53	2
	64	1.18e-09	1.57e-07	0.25	2
	128	2.29e-09	1.57e-07	0.18	2
5	1	1.83e-10	1.42e-07	28.05	2
	2	1.82e-09	7.64e-15	15.16	2
	4	1.36e-09	9.46e-15	4.85	2
	8	4.18e-10	1.42e-07	1.12	2
	16	3.19e-10	1.42e-07	0.62	2
	32	8.12e-09	8.93e-15	0.49	2
	64	3.21e-09	7.51e-15	0.34	2
	128	6.62e-10	1.42e-07	0.61	2
6	1	1.19e-10	1.47e-07	27.36	2
	2	2.86e-10	1.47e-07	7.01	2
	4	1.15e-06	2.08e-12	7.10	2
	8	4.19e-07	1.23e-12	1.72	2
	16	1.72e-10	1.47e-07	1.07	2
	32	2.66e-10	2.39e-15	0.64	2
	64	2.75e-07	3.02e-13	0.23	2
	128	1.46e-08	1.81e-12	0.17	2
7	1	1.16e-10	1.65e-07	26.08	2
	2	1.52e-10	1.02e-14	7.06	2
	4	1.14e-10	8.07e-15	2.99	2
	8	1.56e-10	1.65e-07	1.54	2
	16	1.75e-10	1.65e-07	0.67	2
	32	1.26e-07	8.91e-14	0.46	2
	64	6.49e-08	4.78e-14	0.22	2
	128	5.04e-07	3.68e-14	0.14	2
8	1	7.11e-09	7.44e-15	32.77	2
	2	4.33e-09	2.24e-07	14.63	2
	4	1.66e-09	2.24e-07	4.61	2
	8	1.32e-09	2.24e-07	1.96	2

Instância		parBP			
Número	p	LDE	RMSD	CPU time	#Sol
	16	5.14e-10	2.24e-07	0.31	2
	32	1.56e-10	2.24e-07	0.44	2
	64	1.19e-10	2.24e-07	0.27	2
	128	6.28e-11	2.24e-07	0.09	2
9	1	2.68e-10	1.31e-14	41.8	2
	2	2.60e-10	1.64e-14	16.84	2
	4	6.32e-08	4.04e-14	6.98	2
	8	9.04e-09	1.75e-07	2.22	2
	16	5.94e-09	1.75e-07	1.65	2
	32	2.97e-09	1.75e-07	1.16	2
	64	6.43e-08	7.24e-14	0.33	2
	128	4.03e-08	2.11e-14	0.18	2
10	1	5.16e-10	9.63e-15	44.04	2
	2	1.94e-08	1.25e-13	26.67	2
	4	4.45e-08	1.18e-14	5.21	2
	8	7.98e-09	1.85e-07	2.15	2
	16	6.50e-08	7.56e-14	1.43	2
	32	2.32e-08	7.56e-14	1.16	2
	64	3.65e-09	7.56e-14	0.31	2
	128	3.91e-08	2.33e-14	1.91	2

Tabela 3.2: Comparação do tempo de CPU com diferentes números de processadores

Podemos observar através da Tabela 3.2 que, na maioria das vezes, as execuções em que mais processos são considerados são mais rápidas, e a redução no tempo é maior do que o esperado (com o dobro de processadores, menos da metade do tempo, ver também a Figura 3.3). Além disso, a qualidade das soluções não muda com p: o valor correspondente ao LDE sempre se aproxima de 0 variando, na maioria dos casos, entre 10^{-10} e 10^{-8}. Portanto, esta versão paralela BP funciona de forma eficiente gerando soluções com boa qualidade.

Podemos destacar que em algumas execuções o resultado é invertido, ou seja, uma quantidade maior de processos leva mais tempo para execução do algoritmo. Por exemplo, para a instância 10 (1EZO), a execução com 64 processos tem duração de apenas 0.31 segundos, enquanto que com 128 processos, o tempo é de 1.91 segundos

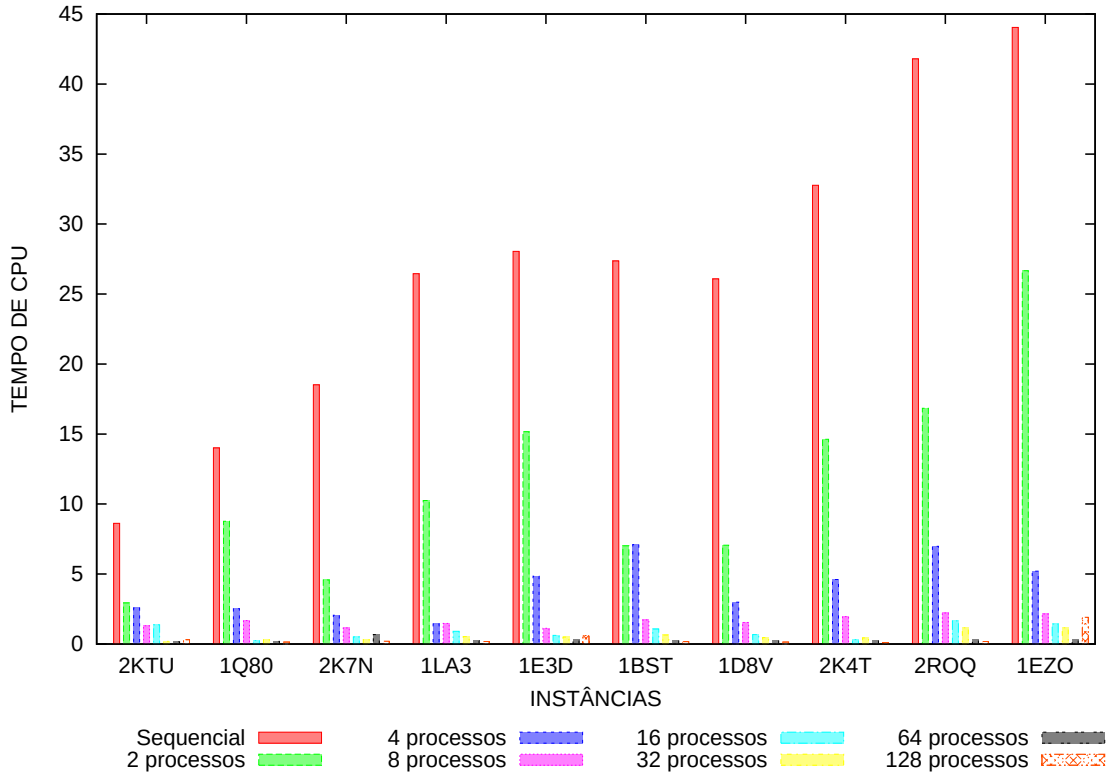

Figura 3.3: Gráfico de comparação do tempo de CPU com diferentes números de processos

(veja Figura 3.4). Neste caso particular, é mais conveniente usar 64 processos, em vez de 128. A conjectura a respeito deste fenômeno é que, provavelmente, devido ao fato da instância original ser subdividida em sub-instâncias cada vez menores, as chamadas locais para o BP terminam mais rapidamente, enquanto a etapa de combinar as soluções locais torna-se mais caro.

Nesta nova versão paralela, diferentemente da versão anterior apresentada em [47], as coordenadas não precisam ser recalculados, mas a viabilidade das soluções ainda precisa ser verificada. Se há muitas distâncias em relação aos átomos previamente atribuídos a diferentes processos, então esta parte do algoritmo paralelo pode se tornar mais cara do que as chamadas para o BP em cada processo. Uma outra conjectura é que o número de processos gerados, em alguns casos, é maior que a quantidade de processadores no ambiente computacional utilizado, o que poderia afetar o tempo computacional, pois, em algum momento, os processos poderiam concorrer por um processador.

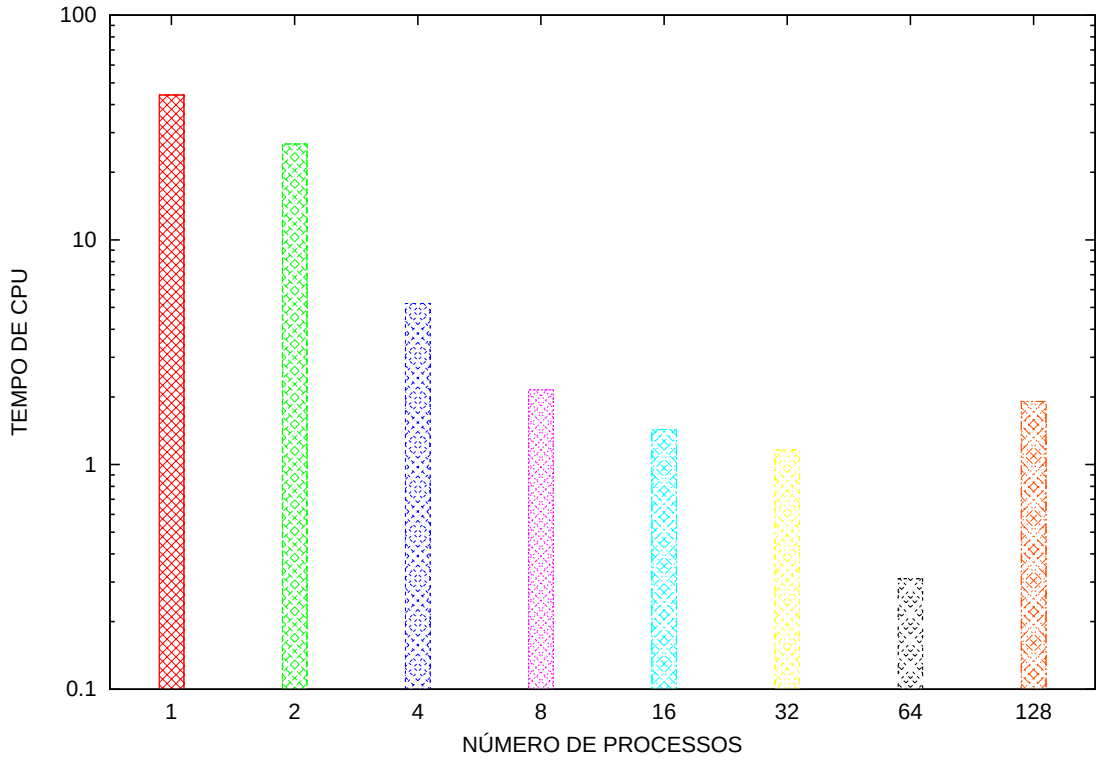

Figura 3.4: Gráfico de comparação do tempo de CPU para instância 10 (1EZO) com diferentes números de processos

Capítulo 4

Problema Discreto de Geometria das Distâncias em Moléculas com Distâncias Intervalares (*i*DMDGP)

Em [33], foi proposto uma nova subclasse do MDGP denominada Interval Discretizable Molecular Distance Geometry Problem (*i*DMDGP), que considera informações de distâncias intervalares, obtidas via experimentos de RMN, e também calculadas a partir de propriedades da molécula de proteínas. Seja $G = (V, E, d)$ um grafo ponderado não direcionado representando uma instância do MDGP. Então, os vértices de G correspondem aos átomos que formam a molécula de proteína e as arestas indicam se as distâncias entre os respectivos átomos são conhecidas ou não. Conforme descrito em [33], considere as seguintes hipóteses que permitem a discretização do problema:

1. O conjunto $E' \subset E$ é formado por todas as distâncias que representam comprimentos de ligações covalentes, juntamente com distâncias entre pares de átomos separados por duas ligações covalentes, que podem ser calculadas a partir dos comprimentos dessas ligações e dos ângulos de ligações. Todas essas distâncias são consideradas valores exatos, uma vez que, distâncias e ângulos de ligações covalentes podem ser considerados como sendo valores fixos em molécula de proteínas [53].

2. Pares de átomos separados por três ligações covalentes, apresentam distâncias não exatas entre si. Dessa maneira, é possível calcular os limites inferior e superior para as distâncias correspondentes, representadas por intervalos, e D distâncias pertencentes a esse intervalo. O conjunto dos pares de átomos que atendem a essa característica é definido como $E'' \subset E$ [49].

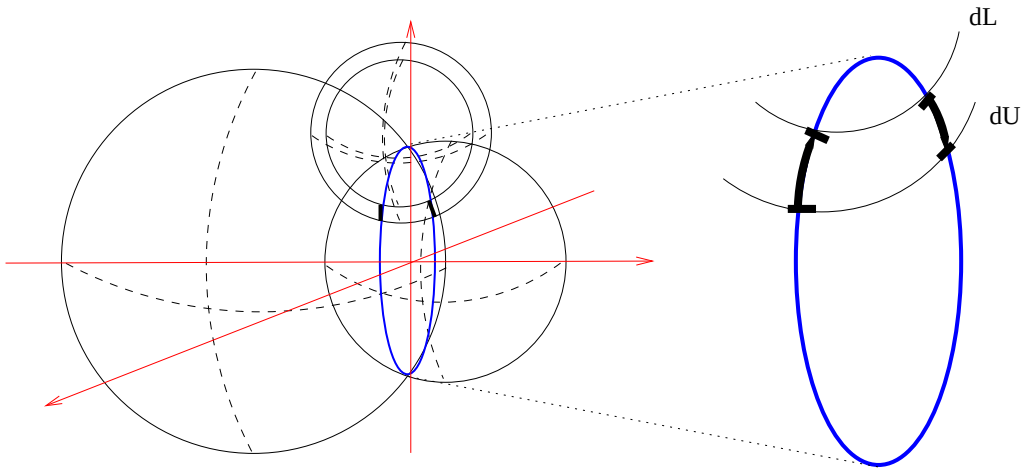

Figura 4.1: Interseção entre duas esferas S_{i-1} e S_{i-2} e concha esférica S^h_{i-3} (Figura retirada de [33]).

3. Existe um conjunto $F \subset E$ das distâncias inter-atômicas que pode ser estimado através da RMN. No entanto, os experimentos de RMN não fornecem as distâncias a todos os possíveis pares de átomos: os átomos devem estar próximos (geralmente, entre 4 e 5Å), sendo, geralmente, átomos de hidrogênio [53].

As Hipóteses 1 e 2 permitem discretizar o problema e descrever um método que utiliza apenas distâncias entre pares de átomos $i, j \in E^{'}$ e $E^{''}$, para discretização do espaço de busca de soluções. As distâncias de RMN (no subconjunto F de E) são usadas apenas para fins de poda (*pruning*). Consequentemente, o novo domínio discreto do problema é completamente independente dos dados experimentais de RMN.

Dada uma ordem em V no iDMDGP, encontra-se a posição de um átomo i considerando a interseção entre duas esferas S_{i-1} e S_{i-2}, e uma concha esférica. Adotando que $S_{i-1} = S(x_{i-1}, d_{i-1}, i)$ e $S_{i-2} = S(x_{i-2}, d_{i-2}, i)$, ou seja, esferas de centros em x_{i-1} e x_{i-2} e raios $d_{i-1,i}$, e $d_{i-2,i}$, respectivamente. Nesse mesmo contexto, tem-se a concha esférica $S^h_{i-3} = S(xi-3, [d^L_{i-3,i}, d^U_{i-3,i}])$, onde seu centro está em x_{i-3} e seu raio pertence ao intervalo $[d^L_{i-3,i}, d^U_{i-3,i}]$. Caso a distância $d_{i-3,i}$ seja um valor exato, S^h_{i-3} também pode ser uma esfera.

A Figura 4.1 destaca a interseção $S_{i-1} \cap S_{i-2} \cap S^h_{i-3}$. No DMDGP, para cada átomo i, considera-se duas posições x_i e $x_i^{'}$. No iDMDGP, caso $d_{k,i}$ seja exata para todo $k \in \{i-3, i-2, i-1\}$, tem-se também duas posições x_i e $x_i^{'}$, no entanto, caso $d_{k,i} \in [d^L_{k,i}, d^U_{k,i}]$ para todo $k = i-3$ tem se $x_i \in [x_i^L, x_i^U]$ e $x_i^{'} \in [x_i^{'L}, x_i^{'U}]$.

Uma definição para o iDMDGP, em termos de grafos, é apresentada a seguir:

Definição 4.1 *Interval Discretizable Molecular Distance Geometry Problem - iDMDGP*
Dado um grafo não direcionado $G = (V, E, d)$*, tal que existe uma ordem* $\{v_1, v_2, \ldots, v_n\} \in V$ *que satisfaça as seguintes condições:*

- $\forall k \in \{4, \ldots, n\}$ *e* $\forall j \in \{k-3, k-2, k-1, k\}$*, com* $i \neq j$*, então* $\{i, j\} \in E$*. Ou seja,* E *contém todas as cliques de quatro vértices consecutivos;*
- *A desigualdade triangular estrita é válida, ou seja,*
- $\forall k \in \{4, \ldots, n\}$ *e* $\forall i\{k-1, k-2\}, \{i, k\} \in E^{'}$*;*
- $\forall k \in \{4, \ldots, n\}$*,* $\{k-3, k\} \in E^{'} \cup E^{''}$*;*

o problema em questão é encontrar $x : V \rightarrow \mathbb{R}^3$ *tal que* $||x_i - x_j|| = d_{ij}$*, para cada* $\{i, j\} \in E$*.*

As condições impostas acima exigem a alteração do algoritmo BP na tentativa de encontrar as soluções desejadas. Faz-se necessária, também, a descrição de uma ordem para V que satisfaça os requisitos da definição do iDMDGP.

4.1 Algoritmo Branch & Prune Intervalar

As limitações de aplicação do algoritmo Branch & Prune em dados reais oriundos de experimentos de RMN, proposto anteriormente em [38], geram a necessidade da definição de uma classe de problemas que considera informações como os comprimentos de ligações covalentes e os ângulos de ligação nas moléculas de proteína. Essa abordagem, denominada iDMDGP, permite a existência de um conjunto de distâncias exatas, bem como um conjunto de distâncias intervalares. Para atender aos requisitos do iDMDGP, foi proposto o algoritmo Branch & Prune Intervalar (iBP) em [33]. O iBP é uma extensão do algoritmo BP clássico.

Um pseudo-código paro o iBP é mostrado no Algoritmo 3, onde, para cada átomo i, podem ocorrer três situações diferentes, dependendo da distância $d(i-3, i)$:

- Se $d(i-3, i) = 0$, o átomo corrente i já apareceu anteriormente na ordem, o que significa que a única posição possível para i é a mesmo que $i-3$;
- Se $d(i-3, i)$ é uma distância exata, aplica-se o BP clássico, e apenas duas posições são possíveis para o átomo i;
- Se $d(i-3, i)$ é uma distância pertencente ao intervalo $[d^L_{i-3,i}, d^D_{i-3,i}]$, escolhe-se D valores no intervalo. Isso produz uma quantidade de $2D$ posições possíveis i.

Note que, para o BP clássico, cada átomo i pode ser posicionado em duas posições gerando uma árvore binária conforme a Figura 2.4, enquanto no iBP, como são escolhidas D distâncias quando se tem $d_{i-3,i} \in [d^L_{i-3,i}, d^D_{i-3,i}]$, obtém-se $2D$ possíveis posições para o nível relacionado ao intervalo. Na Figura 4.2, por exemplo, o nó no nível 7 tem não apenas 2 ramificações como no BP clássico, mas sim, D ramificações para cada nó do nível anterior.

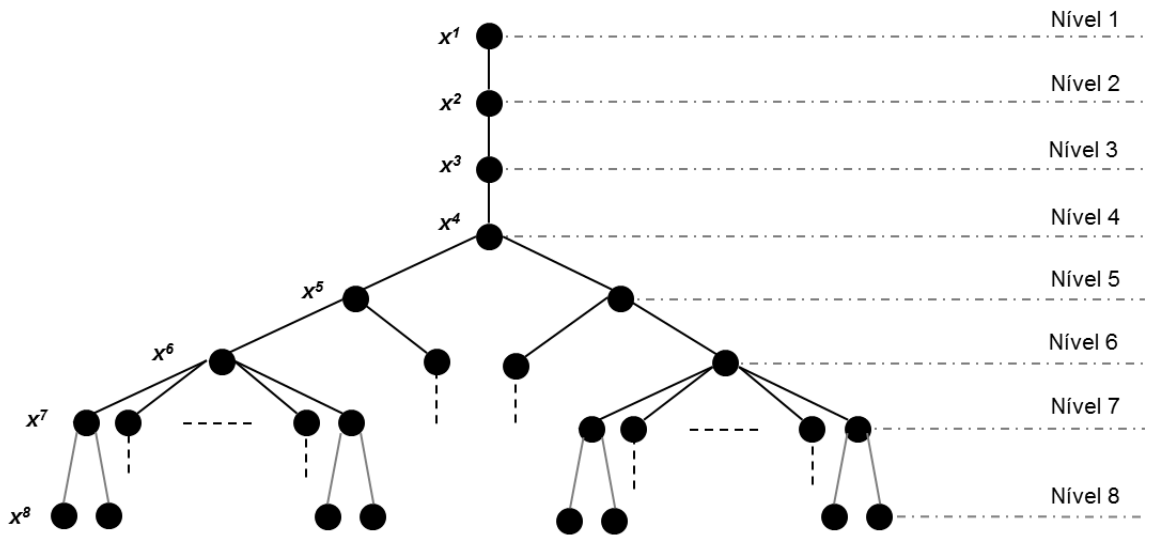

Figura 4.2: Árvore para iBP

Para aplicação do Algoritmo 3, supõe-se a existência de uma ordem para o conjunto V tal que as hipóteses da definição 4.1 sejam satisfeitas. No próximo capitulo será apresentado ordens capazes de satisfazer essas condições.

Algoritmo 3 Algoritmo Branch & Prune Intervalar

1: iBP(j, r, d, D)
2: **if** (r_j é um átomo duplicado) **then**
3: copie as coordenadas do r_j em $x^1_{r_j}$;
4: iBP$(j+1, r, d, D)$;
5: **else**
6: **if** ($d(r_j - 3, r_j)$ é uma distância exata) **then**
7: $b = 2$;
8: **else**
9: $b = 2D$;
10: **end if**
11: **for** $k \in \{1, \ldots, b\}$ **do**
12: calcule a k-ésima posição $x^k_{r_j}$ para o r_j-ésimo átomo;
13: verifique a viabilidade da posição $x^k_{r_j}$ usando distâncias do conjunto F;
14: **if** ($x^k_{r_j}$ é viável) **then**
15: **if** $(j = |r|)$ **then**
16: uma solução x foi encontrada;
17: **else**
18: iBP$(j+1, r, d, D)$;
19: **end if**
20: **end if**
21: **end for**
22: **end if**

Capítulo 5

Ordens Artificias em Moléculas de Proteínas

A principal hipótese para uma instância do iDMDGP está fortemente relacionada com a existência de uma ordem especial para os átomos da molécula. Desta forma, sabe-se que, dado uma instância MDGP que não pode ser discretizada, pode haver uma ordem adequada para os seus átomos para os quais se torna possível a discretização. Nesse sentido, em [33], foi proposta uma forma de ordenar esses átomos de modo a atender as hipóteses de discretização. Com o intuito de facilitar a satisfação dessas hipóteses, alguns átomos podem se repetir ao logo da estrutura. Este problema é escrito como ordem com repetição, como define-se a seguir.

Definição 5.1 *Ordem com Repetição [33].*
Uma ordem com repetição é uma sequência $r : \mathbb{N} \longrightarrow V \cup \{0\}$ *de tamanho* $|r| \in \mathbb{N}$, *tal que:*

- *Os vértices de índices* r_1, r_2, r_3 *formam uma clique.*
- *Para todo* $i \in \{4, \ldots, |r|\}$, *tem-se* $\{r_{i-2}, r_i\}, \{r_{i-1}, r_i\} \in E^{'}$, *que são arestas de* G *que representam distâncias exatas.*
- *Para todo* $i \in \{4, \ldots, |r|\}$, *tem-se que o conjunto* $\{r_{i-3}, r_i\}$ *pode ser unitário, se* $r_{i-3} = r_i$ *(quando ocorre a repetição de um dos vértices de* G*), ou é uma aresta em* $E^{'} \cup E^{''}$.

A ordem com repetição torna possível considerar, para discretização, distâncias que não dependam de dados advindos de experimentos de RMN. Mais especificamente, sobre cada conjunto de três antecessores adjacentes, apenas um está relacionado a uma distância intervalar, sendo que esse intervalo não é advindo de dados de experimentos de RMN, mas sim a uma propriedade de ângulos de torção e, em especial, pode-se calcular limites inferiores e superiores para estes intervalos, como

mencionado em [33]. A ordem com repetição permite a obtenção de cadeias principais artificias de proteínas.

Para a aplicação dessa ordem com repetição em uma proteína, será usada a representação, conforme proposto em [33], onde um aminoácido pode ser representado por um grafo G_{AA}, conforme mostra a Figura 5.1, na qual hidrogênio, nitrogênio, carbono, oxigênio são nomeados, respectivamente, pelas letras H, N, C e O. O carbono ligado a cadeia secundária, presentada por G_{SC}, é escrito como C_α, e H_α trata-se do hidrogênio ligado ao carbono C_α

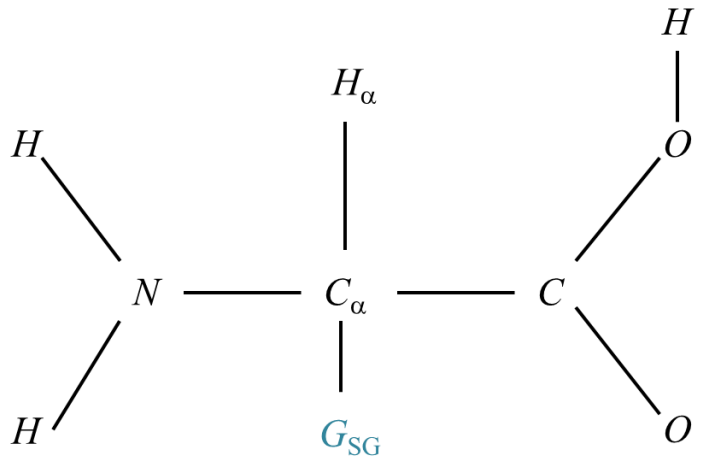

Figura 5.1: Representação de um aminoácido na forma de grafo.

A ligação entre dois aminoácidos também será representada conforme proposto em [33], como mostrada na Figura 5.2, onde o grupo carboxílico ($COOH$) é "resumido" no vértice C^1 e uma ligação $N-H$ do grupo amina ($NH2$) é "resumida" no vértice N^2. Assim, na Figura 5.3, pode observar-se como será a ligação entre p aminoácidos.

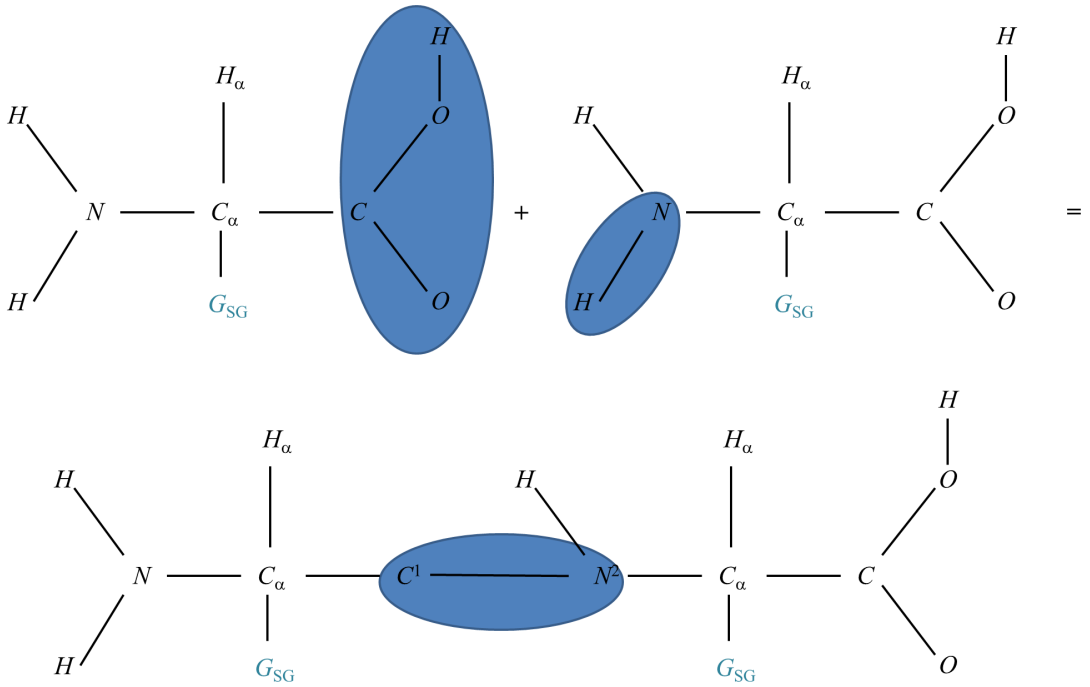

Figura 5.2: Junção entre dois aminoácidos.

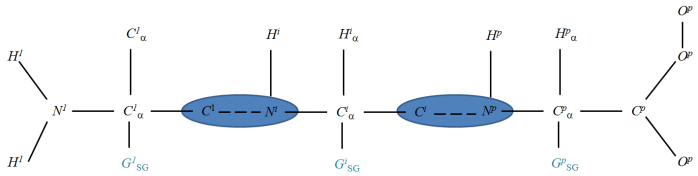

Figura 5.3: Junção de p aminoácidos.

5.1 Uma Ordem para a Cadeia Principal de uma Proteína

Em [33], foi proposto uma ordem especial para os átomos que formam a cadeia principal de uma proteína, que foi provado ser uma ordem com repetição. Essa ordem, consiste no seguinte:

- Para o primeiro aminoácido, considera-se a sequência r^1_{PB}, como mostrado na Figura 5.4:

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

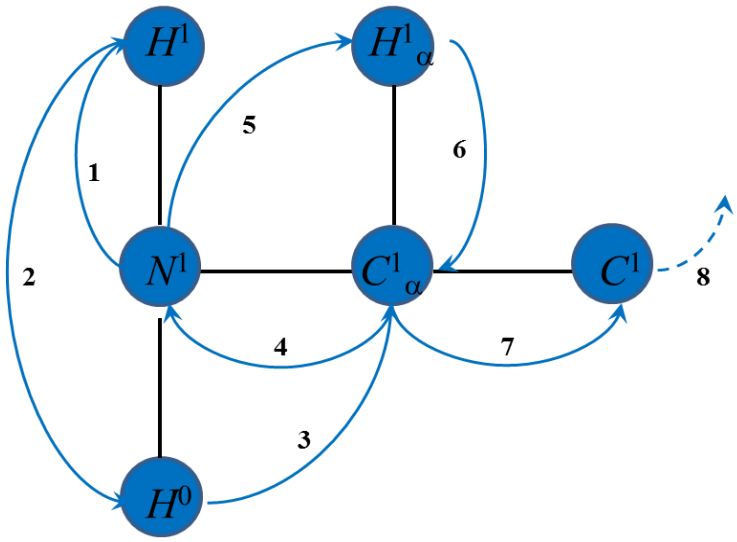

Figura 5.4: Ordem r^1_{PB}.

- Para o segundo aminoácido, considera-se a sequência r^2_{PB}, da forma como visto na Figura 5.5:

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

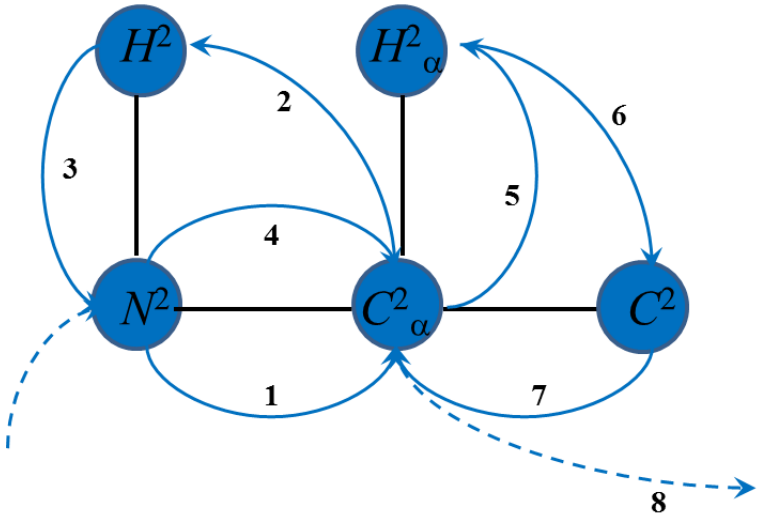

Figura 5.5: Ordem r^2_{PB}.

- Para um aminoácido genérico, considera-se a sequência r^i_{PB}, conforme descrita na Figura 5.6:

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

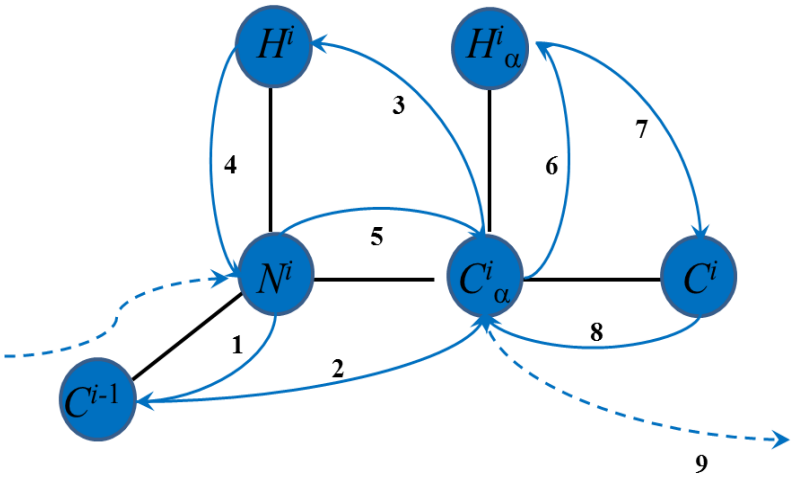

Figura 5.6: Ordem r^i_{PB}.

- Finalmente, para o último aminoácido, considera-se a sequência r^p_{PB}, conforme a Figura 5.7:

$$r^p_{PB} = \{N^p, C^{p-1}, C^p_\alpha, H^p, N^p, C^p_\alpha, H^p_\alpha, C^p, C^p_\alpha, O^p_1, C^p, O^p_2\}$$

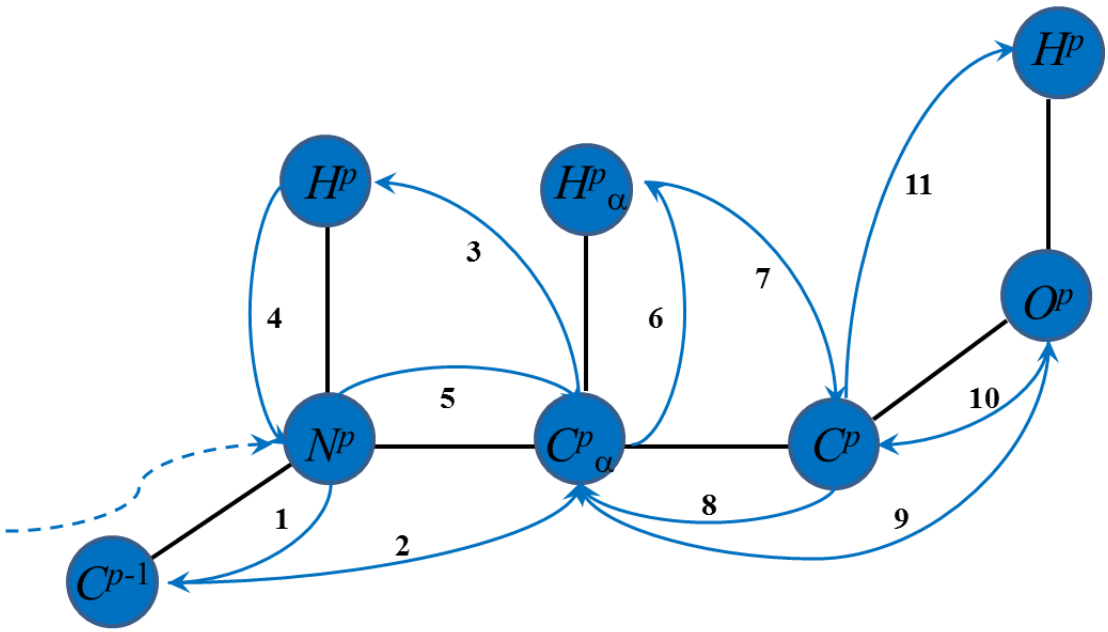

Figura 5.7: Ordem r^p_{PB}.

Sequência	Átomos	Aminoácido
1	N	1
2	H	1
3	H	1
4	C_α	1
5	N*	1
6	H_α	1
7	C_α*	1
8	C	1
9	N	2
10	C_α	2
11	H	2
12	N*	2
13	C_α*	2
14	H_α	2
15	C	2
16	C_α*	2
17	N	3
18	C*	2
19	C_α	3
20	H	3
21	N*	3
22	C_α*	3
23	H_α	3
24	C	3
25	C_α*	3
26	O	3
27	C*	3
28	O	3

Tabela 5.1: Ordenação para a cadeia principal de uma proteína com 3 aminoácidos, conforme ordem mostrada na Figura 5.8. O * indica que o átomo é uma repetição.

Assim, indica-se pelo símbolo r_{PB}, a ordem dos vértices definida para a cadeia principal, sendo que:

$$r_{PB} = \bigcup_{i=1}^{p} rp_{PB}^{i}.$$

A Figura 5.8 mostra uma ordem com repetição para uma cadeia principal de uma proteína pequena contendo 3 aminoácidos. A sequência dos átomos pode ser vista na Tabela 5.1. Nota-se que, no intuito de satisfazer os pressupostos na definição 5.1, alguns átomos são repetidos duas ou três vezes.

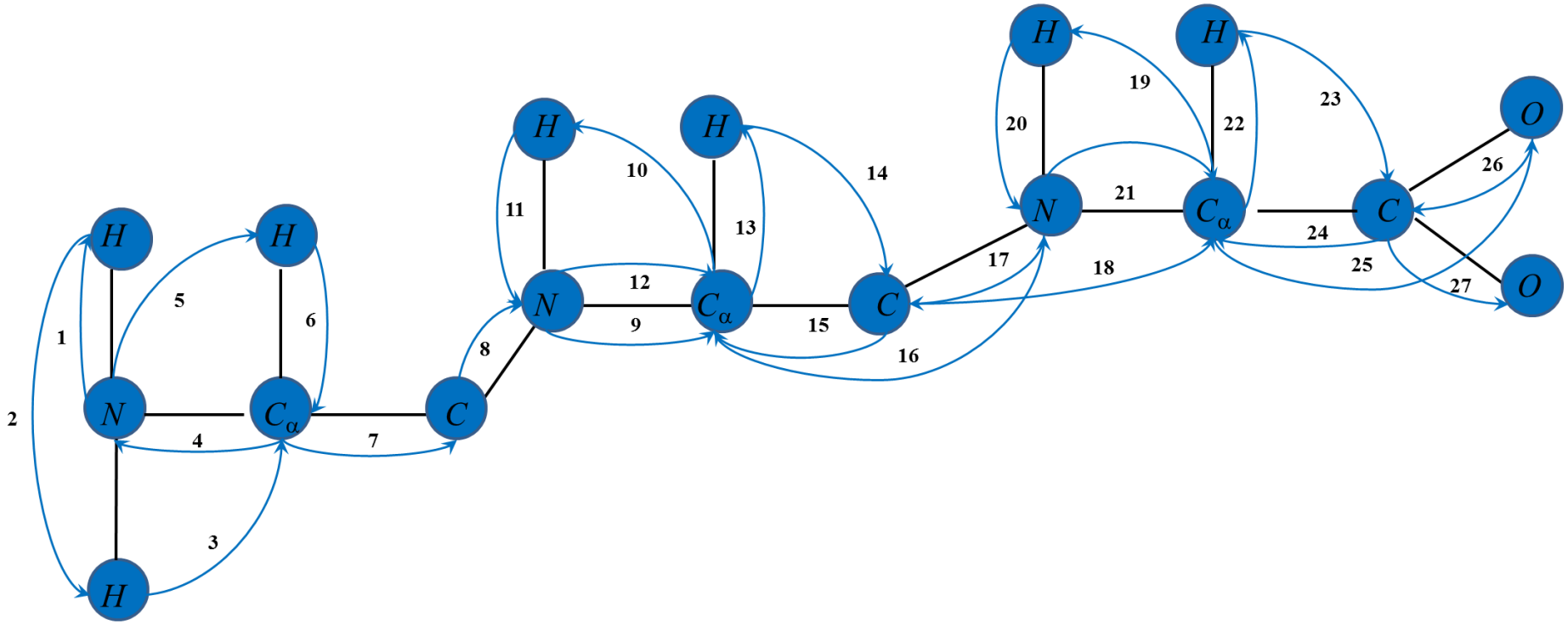

Figura 5.8: Ordem para a cadeia principal de uma proteína.

5.2 Nova Ordem para a Cadeia Principal de uma Proteína

Nesta seção, propomos uma nova ordem especial para os átomos que formam a cadeia principal de uma proteína. Essa ordem, consiste no seguinte:

- Para o primeiro aminoácido, considera-se a sequência r^1_{PB}, conforme Figura 5.9:

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

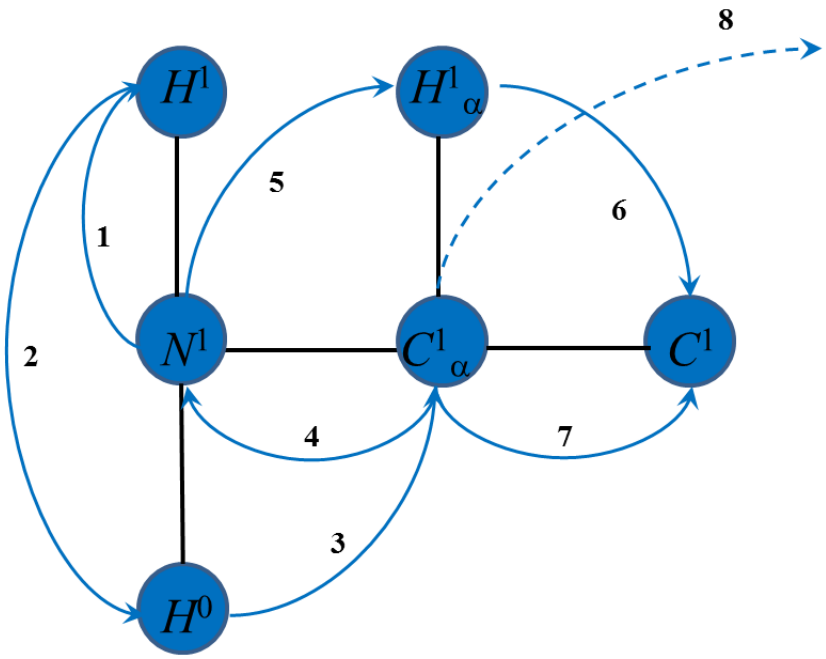

Figura 5.9: Nova ordem para r^1_{PB}.

- Para o segundo aminoácido, conforme Figura 5.10, considera-se a sequência

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

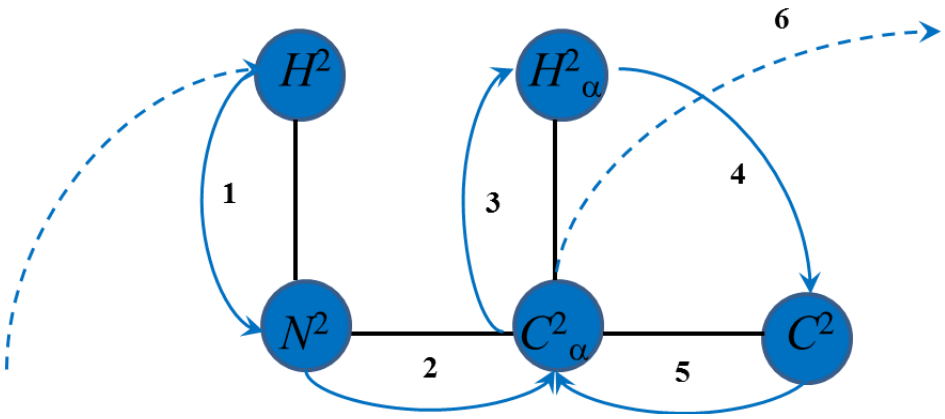

Figura 5.10: Nova ordem para r^2_{PB}.

- Para um aminoácido genérico, conforme Figura 5.11, tem-se

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

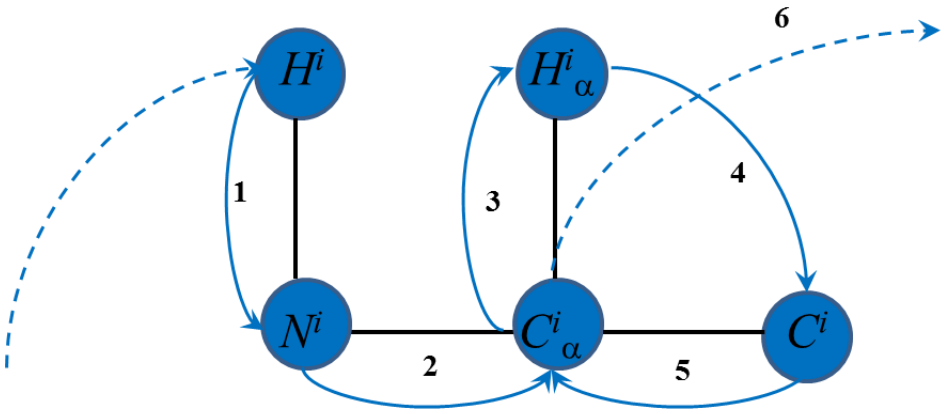

Figura 5.11: Nova ordem para r^i_{PB}.

- Finalmente, para o último aminoácido, considera-se a sequência conforme Figura 5.12:

$$r^p_{PB} = \{H^p, N^p, C^p_\alpha, H^p_\alpha, C^p, C^p_\alpha, O^p_1, C^p, O^p_2\}$$

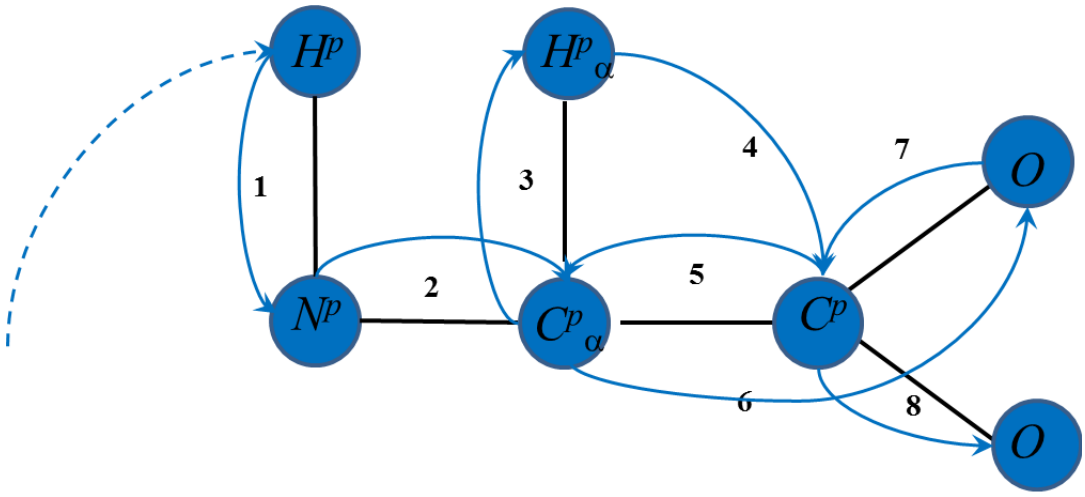

Figura 5.12: Nova ordem para r^p_{PB}.

A Figura 5.13 mostra a nova ordem proposta para a cadeia principal de uma proteína pequena contendo 3 aminoácidos.

Na próxima seção, apresentaremos comparações entre a ordem apresentada na seção anterior e a nova ordem proposta.

5.3 Comparações entre as ordens

Consideremos, conforme Figuras 5.8 e 5.13, uma cadeia principal de uma molécula de proteína contendo 3 aminoácidos. Podemos observar que, para discretizar o problema associado à cadeia principal de proteína, 28 vérticies são necessários para a ordem na Figura 5.8, considerando a repetição, enquanto que a ordem definida na Figura 5.13 consiste apenas de 23 vértices. Além disso, a ordem na Figura 5.13 explora algumas restrições químicas para considerar algumas distâncias exatas. Por exemplo, a distância entre o átomo C_α e o átomo H, pertencente ao aminoácido seguinte, pode ser calculada como um valor exato, por causa da ligação peptídica entre dois aminoácidos consecutivos. Esta distância é explorada na nova ordem.

Uma instância pequena, com 3 aminoácidos, foi construída para cada uma das ordens mostrada nas Figuras 5.8 e 5.13, conforme definida em [33]: Dado quatro números d_1, d_2, l_3, u_3, tais que $2d_1 > d_2$ e $l_3 < u_3$, a distância entre um par $\{u, v\}$ de átomos ligados entre si é $d_{u,v} = d_1$, a distância entre um par de átomos $\{u, v\}$ separados por duas ligações covalentes é $d_{u,v} = d_2$. Para cada par de átomos $\{u, v\}$ separados por três ligações covalentes, é gerado um conjunto discreto contendo D distâncias no intervalo $[l_3, u_3]$.Pode observar-se que a instância não contém nenhuma informação relacionada ao conjunto F, isto é, não haverá *pruning* para o iBP. Deste modo, o iBP gera uma árvore completa e então escolhe, aleatoriamente, um vértice

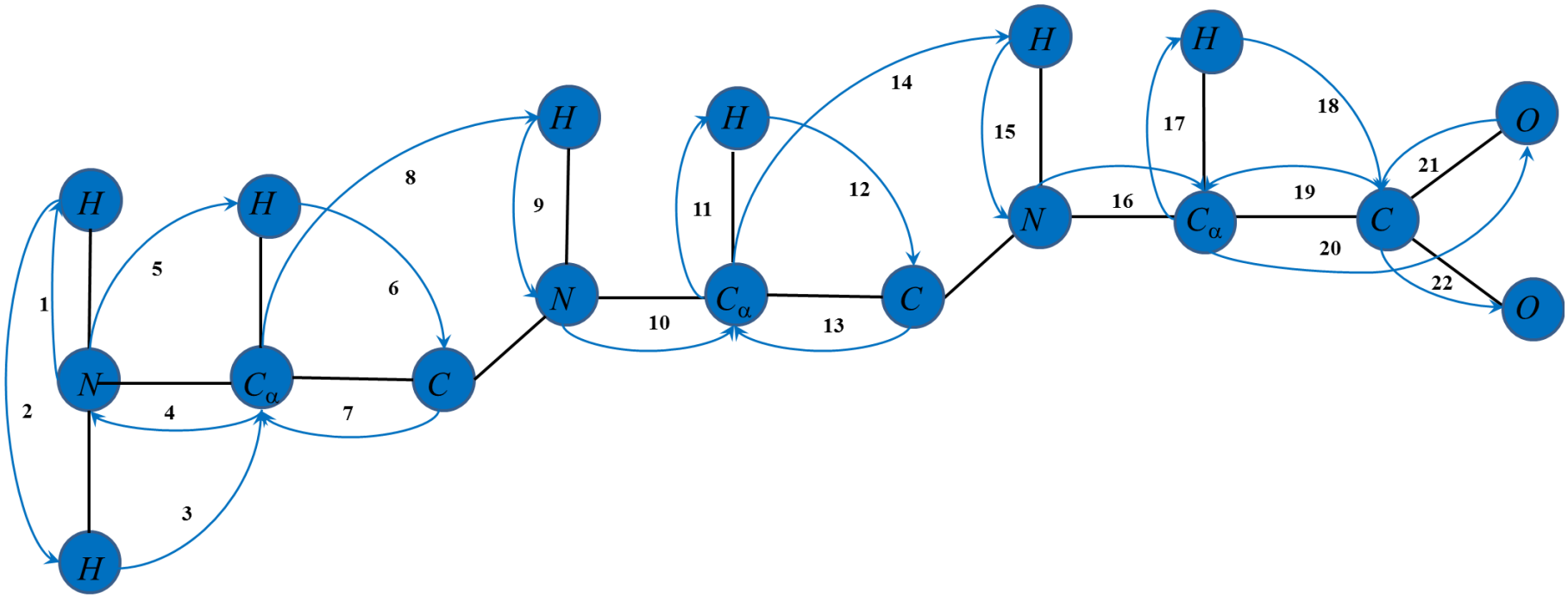

Figura 5.13: Nova ordem para cadeia principal de uma proteína

Sequência	Átomos	Aminoácido
1	N	1
2	H	1
3	H	1
4	C_α	1
5	N*	1
6	H_α	1
7	C	1
8	C_α*	1
9	H	2
10	N	2
11	C_α	2
12	H_α	2
13	C	2
14	C_α	2
15	H	3
16	N	3
17	C_α	3
18	H_α	3
19	C	3
20	C_α	3
23	O	3
24	C*	3
25	O	3

Tabela 5.2: Nova ordenação para a cadeia principal de uma proteína com 3 aminoácidos, conforme Figura 5.13.

no nível $|r|$ e calcula todas as distâncias menores que 5Åpara formar o conjunto F. Para cada aresta $\{u, v\} \in F$, atribui-se o intervalo $[d_{u,v} - \epsilon, d_{u,v} + \epsilon]$.

A Figura 5.14 mostra a estrutura da árvore relacionada com a instância com 3 aminoácidos. As posições dos primeiros três átomos podem ser obtidas utilizando a informação conhecida sobre as distâncias em $E^{'}$. A ramificação da árvore começa no nível 4, que corresponde ao átomo de C^1_α. No entanto, devido à propriedade de simetria do DMDGP, podemos descartar um dos ramos presentes no nível 4, e concentrar a busca em apenas um deles. No nível 5, temos o primeiro átomo repetido, o nitrogênio N^1*, que já apareceu no nível 1. Assim, não temos nenhuma ramificação, porque a nova cópia de N^1 só pode ser colocada na mesma posição do N^1 original. Já no nível 6, aparece o primeiro hidrogênio sobre o qual é necessário ramificar. Devido ao fato de a distância entre este átomo e o H^1 anterior ser uma distância intervalar, é preciso discretizá-la em d distâncias exatas. Como consequência, os ramos $2D$ são adicionados no nível 6 na árvore binária. No nível 7, encontra-se outro átomo repetido e, portanto, não há nenhuma ramificação. Após este átomo, dispõe-se de uma sequência de três átomos que não são nem repetidos e nem hidrogênios, assim 2 ou $2D$ ramos são adicionados à árvore. O primeiro hidrogênio do segundo aminoácido está no nível 11, uma vez que a distância entre C^1 e H^2 pertence a $E^{'}$, que tem apenas dois ramos. Os próximos níveis são semelhantes aos anteriores.

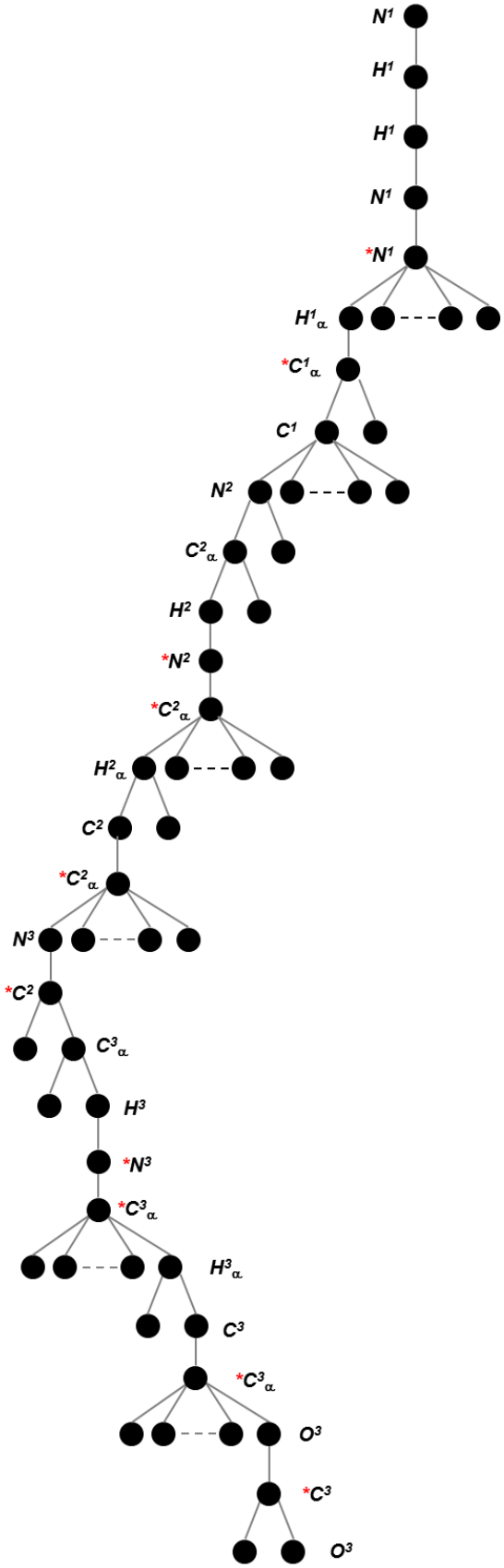

Figura 5.14: Parte da árvore para uma instância com 3 aminoácido, utilizando a ordem definida na seção 5.1, conforme a Figura 5.8.

Nesse contexto, a Figura 5.15 apresenta a estrutura da árvore para a ordem proposta na Figura 5.13. As configurações dos átomos até o nível 6 são equivalentes ao demonstrado na Figura 5.14. Já no nível 7, adiciona 2 ramos, enquanto no nível 8, esse processo não ocorre, uma vez que identifica-se a presença de outro átomo repetido. No nível 9, devido à distância entre H^2 e H^1_α ser uma distância intervalar, faz-se necessário a discretização em d distâncias. Observa-se que, neste nível, o intervalo ocorre entre átomos de H, enquanto que na Figura 5.8 ocorre entre $H-N$. É importante destacar que a distância entre átomos de hidrogênios, próximos é conhecida (por exemplo, via RMN), permitindo, assim, a poda e consequente redução do número de ramos para o próximo nível. Note que, enquanto na Figura 5.8 as distância intervalares podem estar relacionadas à ligações do tipo $H - N$, na Figura 5.13 essas relações ocorrem apenas entre átomos de H. Esse fato pode ser averiguado na Tabela 5.3, que apresenta a comparação entre as duas ordens, mostrando o número de ramificações em cada nível das árvores das Figuras 5.14 e 5.15, respectivamente, uma vez que no átomo 9 pode-se constatar a presença de 240 ramificações para a ordem proposta na Figura 5.8, e um total de 74 ramificações para a ordem proposta na Figura 5.13. Situações semelhantes são observadas em outros níveis.

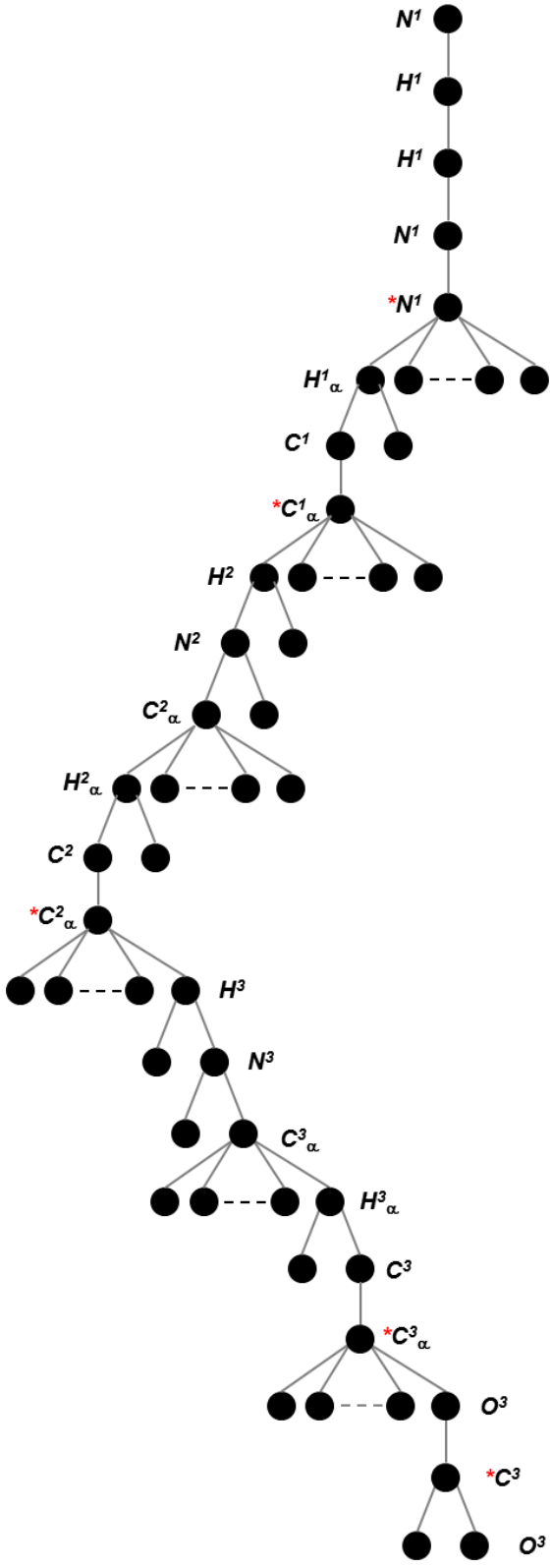

Figura 5.15: Parte da árvore para uma instância com 3 aminoácido, utilizando a nova ordem definida na seção 5.2, conforme a Figura 5.13.

A Tabela 5.4 mostra algumas experimentos computacionais. Todas as instâncias

Ordem conforme Figura 5.8			Nova Ordem conforme Figura 5.13		
Sequência	Átomo	Número de ramificações	Sequência	Átomo	Número de ramificações
1	N	1	1	N	1
2	H	1	2	H	1
3	H	1	3	H	1
4	C_α	2	4	C_α	2
5	N*	2	5	N*	2
6	H_α	12	6	H_α	12
7	C_α*	12	7	C	24
8	C	24	8	C_α*	24
9	**N**	**240**			
10	C_α	240			
11	H	84	9	H	78
12	N*	84	10	N	78
13	C_α*	84	11	C_α	78
14	H_α	18	12	H_α	22
15	C	36	13	C	44
16	C_α*	36	14	C_α*	44
17	**N**	**360**			
18	C*	360			
19	C_α	360			
20	H	10	15	H	28
21	N*	10	16	N	28
22	C_α*	10	17	C_α	28
23	H_α	10	18	H_α	4
24	C	10	19	C	8
25	C_α*	10	20	C_α*	8

Tabela 5.3: Comparação entre as duas ordens

foram geradas artificialmente. Para cada instância, (n_{aa}) indica o número de aminoácidos que compõem a instância, $|E|$ é a cardinalidade do conjunto de distâncias em E, $\min(D)$ é o valor mínimo para D tal que iBP é capaz de encontrar pelo menos uma solução, #Sol é o número total de soluções, e o tempo de CPU é dado em segundos. Todos os experimentos foram realizados em um computador portátil. Podemos observar que o número de pontos de discretização D para os intervalos é sempre menor quando a nova ordem é considerada. O número total de soluções obtidas, bem como a eficácia das execuções, também dependem da ordem.

	Ordem conforme Fig. 5.8				Ordem conforme Fig. 5.13			
n_{aa}	$\|E\|$	$\min(D)$	#Sol	tempo de CPU	$\|E\|$	$\min(D)$	#Sol	tempo de CPU
3	80	4	912	0.01	63	3	224	0.00
4	112	4	456	0.02	84	3	736	0.01
5	143	4	5472	0.02	104	3	8096	0.01
6	176	4	60	0.13	126	3	188	0.03

Tabela 5.4: A comparação entre as ordens para instâncias pequenas.

Capítulo 6

Ordens válidas para o DDGP

Instâncias do DDGP podem ser resolvidas através do algoritmo Branch & Prune (BP) [29] que é, potencialmente, capaz de enumerar um conjunto completo de soluções. Esta é a principal diferença entre o algoritmo BP e outros algoritmos para o DGP. No entanto, para se aplicar o BP, as hipóteses do DDGP devem ser satisfeitas. Assim, encontrar uma ordem para os vértices de V, tal que estas hipóteses sejam satisfeitas, representa uma importante etapa de pré-processamento para a solução de DDGPs [48]. Dizemos que uma ordem para os vértices de V é valida para o DDGP se ela satisfaz as hipóteses do DDGP (ver Def. 2.1). Este problema é referido como Discretizing Vertex Order Problem (DVOP) [34].

Seja $G = (V, E, d)$ um grafo ponderado não-direcionado relacionado a uma instância DGP e suponha que uma ordem total está associada aos vértices em V (sabe-se que, a partir de qualquer ordem parcial em V, uma ordem total pode ser derivada). Para se referir a uma ordem, consideramos o símbolo $<$ e associamos índices quando for necessário fazer a distinção entre diferentes ordens (por exemplo, $<_1$ ou $<_2$). Da mesma forma, o símbolo $(u, v)_{<_1}$ irá se referir a uma aresta envolvendo os vértices u e v na ordem $<_1$. Consideramos a uma ordem $<$ para as quais as hipóteses na Def. 2.1 são satisfeitas, em dimensão K, como um DDGP K-ordem.

Seja $\alpha_<(v)$, para $v \in V$, o número de antecessores adjacentes de v na ordem $<$, tal que:

$$\alpha_<(v) = card\{u \in v : (u, v)_< \in E\}.$$

Do mesmo modo, seja $\beta_<(v)$, para $v \in v$, o número de sucessores adjacentes de v, na ordem $<$:

$$\beta_<(v) = card\{u \in v : (v, u)_< \in E\}.$$

Definição 6.1 *The Discretizing Vertex Order Problem (DVOP)*
Dado um grafo não direcionado $G = (V, E)$ e um inteiro positivo K, verificar se existe uma ordem $<$ em V tal que:

(a) os primeiros vértices na ordem formam uma K-clique, e

(b) para cada $v \in V$, $\alpha_<(v) \geq K$.

Observe que o DVOP não verifica se a ordem satisfaz a hipótese de determinante de Cayley-Menger, dada na Def. 2.1. Isso ocorre porque a possibilidade do conjunto de matrizes de distâncias produzir um determinante de Cayley-Menger com valor zero tem medida de Lebesgue igual a zero dentro do conjunto de todas as possíveis matrizes de distâncias (reais) [34]. A probabilidade de isso acontecer é, portanto, 0 em um sentido matemático. A NP-completude do DVOP segue trivialmente da NP-completude do problema K-clique, pois encontrar um DDGP K-ordem implica em determinar K vértices formando uma clique em G. Quando K é fixo, no entanto, como em aplicações reais, o DVOP pode ser resolvido em tempo polinomial [34]. No entanto, é importante ressaltar que pode não existir um DDGP K-ordem, conforme proposição 6.1.

Proposição 6.1 *Dado um grafo não direcionado ponderado* $G = (V, E, d)$ *e uma ordem* $<$ *em* V*, não existe DDGP* K*-ordens se algum vértice tem grau menor que* K*.*

Ressalta-se que, no caso de existir vértices para os quais a soma de $\alpha_<(v)$ e $\beta_<(v)$ é menor que K, pode-se remover este subconjunto de vértices e trabalhar no subgrafo correspondente. Note que a Proposição 6.1 não pode ser invertida, ou seja, pode existir instâncias que não admitem qualquer DDGP K-ordem, mesmo que, para todo $v \in V$, $\alpha_<(v) + \beta_<(v) \geq K$.

6.1 Algoritmo Guloso para o DVOP

O algoritmo guloso, apresentado nesta seção, foi proposto em [34] para o DVOP e testados em instâncias baseadas em proteínas. A idéia básica do algoritmo (ver pseudocódigo em Alg. 4) é encontrar uma K-clique C em G e considerar os vértices em C no início da nova ordem. Desta forma, a hipótese (a) do DVOP é satisfeita (veja Def. 6.1). Em seguida, todos os outros vértices são posicionados na nova ordem, procurando por aqueles com o maior número de vértices adjacentes. Se o número de vértices adjacentes é sempre maior ou igual a K, para uma determinada clique C inicial, então, uma ordem satisfazendo ambas hipóteses (a) e (b) existe. Caso contrário, se para todos as cliques possíveis C, existe pelo menos um vértice para o qual o número de vértices adjacentes é menor do que K, então, uma ordem satisfazendo a hipótese (b) não existe. Mais detalhes sobre este algoritmo pode ser encontrado em [34].

Este algoritmo guloso é polinomial, pois a enumeração de todos as K-cliques em G pode ser realizada em tempo polinomial, bem como a construção de cada ordem,

Algoritmo 4 Algoritmo guloso para encontrar ordem válida para o DDGP

```
1: reordenar(G)
2: while DDGP K-ordem não é encontrada do
3:    encontre uma K-clique C em G;
4:    repace os vértices de C para o inicio da nova ordem: B = C;
5:    while V ∖ B ≠ ∅ do
6:       encontre o vértice v ∈ V ∖ B com o maior número l de vértices adjacentes
         em B;
7:       if l < K then
8:          break o laço while: não existe ordenação possível para esta escolha de
            C;
9:       end if
10:      B = B + {v};
11:   end while
12: end while
13: ordem < definida por B.
```

selecionando os vértices com o maior número de vértices adjacentes. Implementamos uma estratégia eficiente para encontrar o vértice com o número máximo de adjacentes, exigido na linha 6 de Alg. 4. Assim, esta implementação do algoritmo guloso é mais eficiente do que o considerado nos experimentos em [34]. Isso garante uma comparação justa com o algoritmo proposto na seção 6.2.

6.2 Novo Algoritmo para o DVOP

Considere que uma ordem $<_1$ para os vértices em G já está disponível. Suponha que esta ordem não é um DDGP K-ordem, e, para cada $v \in V$, $\alpha_{<_1}(v) + \beta_{<_1}(v) \geq K$, para garantir que essa ordem pode existir. A ideia básica deste algoritmo é selecionar todo v para que $\alpha_{<_1}(v) < K$, e modificar a sua posição de modo que, na nova ordem $<_2$, tenha-se que $\alpha_{<_2}(v) = K$ e $\beta_{<_2}(v) = \beta_{<_1}(v) - \alpha_{<_1}(v)$.

Ao considerar a ordem $<_1$, suponha que v' é tal que $\alpha{<_1}(v') < K$. Seja $h = \beta_{<_1}(v') - \alpha_{<_1}(v')$ e $\Xi = \{u \in V : (v', u)_{<_1} \in E\}$. A partir da ordem de $<_1$, uma ordem nos vértices de Ξ pode ser obtida, de modo que o $h^{\textbf{ésimo}}$ elemento pode ser selecionado, por exemplo v''. Nesta nova ordem $<_2$, move-se v' logo após v'', o que implica que $\alpha_{<_2}(v') = K$. Os vértices entre a antiga e a nova posição para v' podem ser afetados por esta mudança, ao passo que a situação permanece inalterada para todos os outros.

Se um vértice v está entre a antiga e a nova posição para v', então o valor de $\alpha_{<_1}(v)$ pode diminuir. Em tal caso, a posição do vértice da ordem precisa de ser modificada, e isto pode ser feito simplesmente aplicando o procedimento acima para os vértices seguintes na posição antiga de v' na ordem de $<_1$.

O pseudocódigo do novo algoritmo para o DVOP pode ser visto no Alg. 5. Este algoritmo requer uma ordem $<_1$ como entrada e, como uma consequência, o desempenho deste algoritmo é dependente da dada ordem inicial.

Algoritmo 5 Novo algoritmo para encontrar ordem válida para o DDGP

```
1: reordenar(G, <_1)
2: cópia ordem <_1 em <_2;
3: define o conjunto B tal que cada v ∈ V está na ordem <_2
4: for cada v ∈ B, na ordem <_2 do
5:    if α_{<_2}(v) < K then
6:       seja Ξ = {u ∈ V : (v, u)_{<_2} ∈ E};
7:       seja h = β_{<_2(v)} − α_{<_2}(v);
8:       seja w = h^ésimo elemento, na ordem <_2, em Ξ;
9:       mova, no conjunto B, v após w;
10:      atualize ordem <_2 (de B atualizado);
11:   end if
12: end for
```

Destaca-se ainda que este algoritmo poderia entrar em ciclos infinitos. Quando há um subconjunto de vértices que são selecionados em repetição, quer dizer que eles formam um subconjunto de vértices que tem menos do que K ligações com o resto. Quando o algoritmo entrar em ciclos, pode-se parar a execução, e o DDGP K-ordem pode não existir.

6.3 Aplicação do DDGP em Redes de Sensores

Com o objetivo de mostrar que o DDGP pode ser aplicado a outros problemas que não estejam relacionados a moléculas de proteínas e considerar instâncias grandes e realistas, simulamos situações do Problema de Localização em Rede Sensores (ou, como na definição em inglês, The Sensor Network Localization Problem) que é definido a seguir.

Definição 6.2 *The Sensor Network Localization Problem (SNLP) [27]*
Consiste na localização da posição de n sensores, $p_i \in \mathbb{R}^r$, $i = 1, \ldots, n$, dado apenas as distâncias Euclidiana $D_{ij} = ||p_i - p_j||_2^2$ entre sensores com raio de comunicação R (medida de potência usada para comunicação), $R > 0$, e dadas as posições de um subconjunto de sensores, $p_i = n - m + 1, \ldots, n$ (chamados âncoras); r representa a dimensão do problema.

Pela Definição 6.2, observa-se que podem existir instâncias do SNLP que não satisfaçam as hipóteses do DDGP (veja, Def.2.1). Entretanto, como mostrado, anteriormente, podemos utilizar algum dos algoritmos apresentados na seções 6.1 e 6.2 para encontrar ordens que transformem instâncias do SNLP em instâncias do

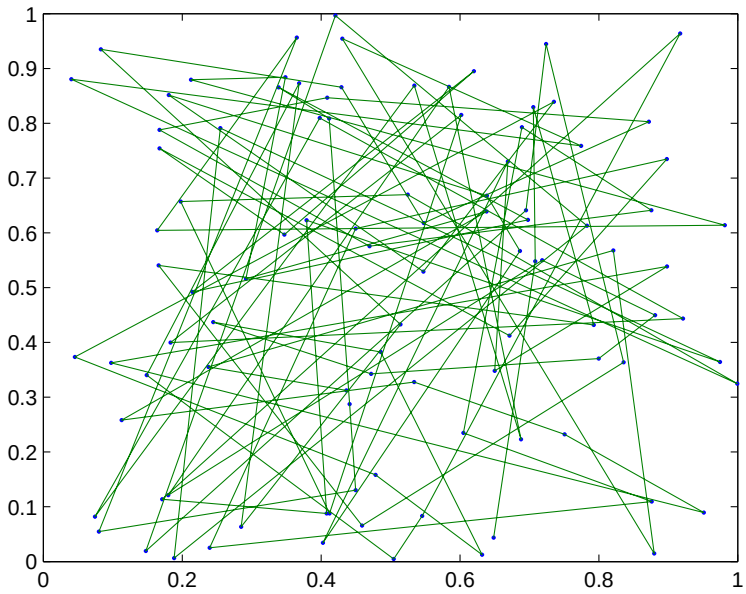

Figura 6.1: Original ordem de uma instância do WSNL com 100 sensores.

DDGP e assim utilizar o algoritmo BP para resolver o SNLP. Na próxima seção, serão mostrados resultados computacionais para os algoritmos para o DVOP.

6.3.1 Resultados Computacionais

Apresentamos nesta seção algumas comparações entre os dois algoritmos considerados nas seções 6.1 e 6.2, respectivamente, aplicados a instâncias do SNLP. Os experimentos foram realizados em um computador Intel(R) Core(TM) i3-2120 CPU@3.30GHz com 8GB RAM. Todos os códigos foram escritos em linguagem de programação C e compilados com o compilador GNU C, versão 4.7.1, sob Linux.

Para estes experimentos, supõe-se que $r = 2$ e que todas as distâncias são conhecidas com valores exatos. Para gerar tais instâncias, empregamos a mesma técnica descrita em [27]: pontos pertencentes a área de $[0, 1]^r$, todas as distâncias entre os pontos, distribuídos aleatoriamente, são menores que um raio de comunicação R predefinido e são conhecidas.

A Figura 6.1 mostra uma instância pequena, com 100 sensores, para o SNLP. Em seguida, a Figura 6.2 e a Figura 6.3 mostram respectivamente, ordens encontradas pelos Algoritmo 4 e Algoritmo 5.

A Tabela 6.1 mostra alguns experimentos computacionais para diferentes tamanhos de n e diferentes valores de raio R. Pode ser observado que o Alg 4 é fortemente dependente do tamanho de n e da cardinalidade de E, pois os experimentos computacionais são mais caros quando os valores de n e $|E|$ são maiores. Por outro lado, o

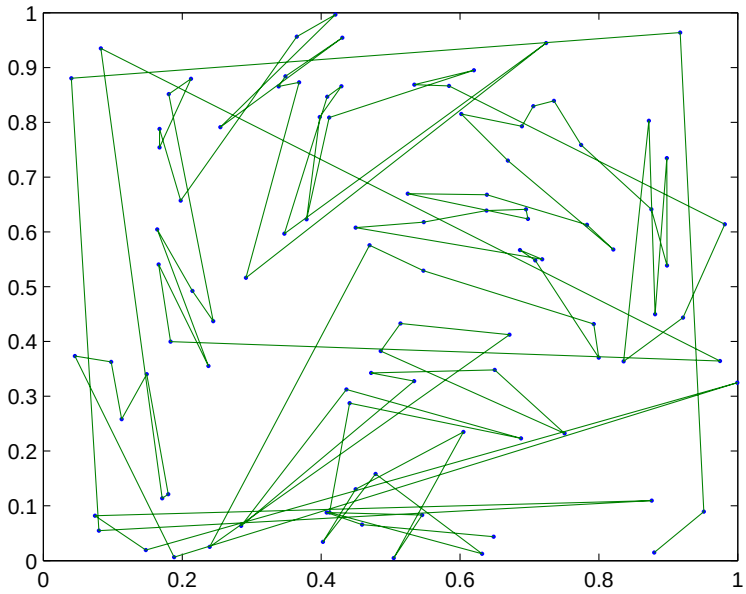

Figura 6.2: Ordem encontrada pelo Alg. 4 para uma instância do WSNL com 100 sensores.

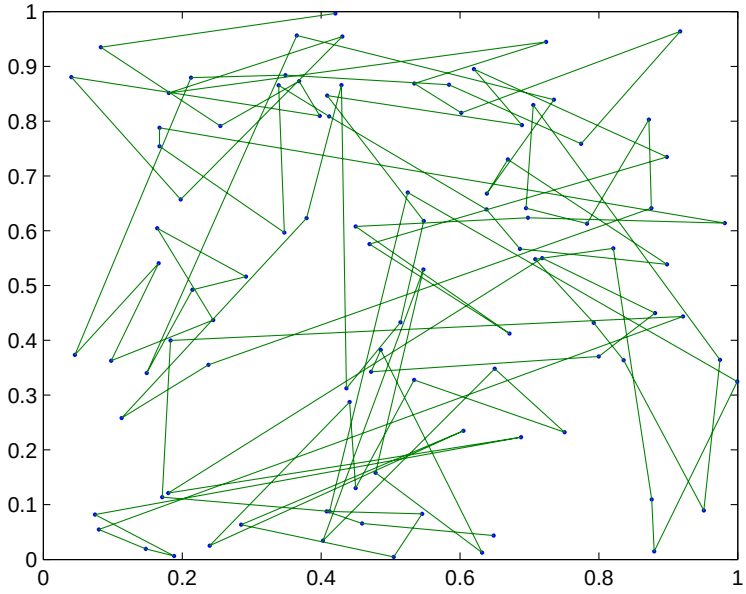

Figura 6.3: Ordem encontrada pelo Alg. 5 para uma instância do WSNL com 100 sensores.

Alg 5 mostra que este comportamento está relacionado apenas com o n, ao mesmo tempo que melhora o seu desempenho quando $|E|$ é maior. A conjectura a respeito desta diferença é que enquanto a identificação do vértice que tem o maior número de antecessores adjacentes (veja Alg 4, linha 6.) é mais caro quando $|E|$ é maior, a presença de mais arestas em E faz com que o Alg. 5 mova menos vértices antes de terminar o "loop" principal.

Mesmo que este novo algoritmo entre em ciclos, em teoria, pode-se destacar que isso nunca aconteceu para as instâncias consideradas do SNLP.

Instâncias			Alg. 4	Alg. 5
n	R	$\|E\|$	CPU time	CPU time
4000	0.05	60351	0.98	1.23
4000	0.06	85815	1.13	0.76
4000	0.07	115511	1.33	0.47
4000	0.08	149606	1.62	0.32
4000	0.09	187789	1.91	0.25
4000	0.10	230116	2.29	0.19
6000	0.05	136532	2.64	1.71
6000	0.06	195323	3.39	0.98
6000	0.07	262742	4.24	0.73
6000	0.08	337476	5.07	0.40
6000	0.09	426764	5.27	0.32
6000	0.10	518907	7.48	0.29
8000	0.05	241590	5.83	3.35
8000	0.06	343873	7.59	1.21
8000	0.07	466346	9.71	1.07
8000	0.08	601909	11.96	0.57
8000	0.09	750550	14.71	0.43
8000	0.10	918520	17.01	0.36
10000	0.05	378545	11.09	3.77
10000	0.06	536711	14.61	2.70
10000	0.07	723071	17.81	0.97
10000	0.08	936524	22.18	0.68
10000	0.09	1182242	27.5	0.44
10000	0.10	1440175	32.69	0.49
15000	0.05	847805	33.55	10.04
15000	0.06	1212149	44.49	3.92
15000	0.07	1627939	57.38	3.17
15000	0.08	2111858	71.9	1.61
15000	0.09	2655316	88.31	1.32
15000	0.10	3232900	106.37	1.11
20000	0.05	1505440	75.8	13.78
20000	0.06	2147869	101.6	8.27
20000	0.07	2903067	133.42	4.68
20000	0.08	3757364	167.36	3.02
20000	0.09	4716007	216.06	2.40
20000	0.10	5770198	250.96	2.02

Tabela 6.1: Comparação entre o Algoritmo. 4 e o Algoritmo. 4 em um conjunto de instância do WSNL.

Capítulo 7

Conclusão e Trabalhos Futuros

Em suma, apresentamos nesta tese alguns algoritmos para resolver e/ou auxiliar na resolução do *Distance Geometry Problem* (DGP).

Primeiramente, este trabalho abordou a formulação discreta do DGP, conhecida como *Discretizable Distance Geometry Problem* (DDGP), considerando conjuntos esparsos de distâncias com valores exatos. Realizamos experimentos computacionais com dados reais obtidos no *Protein Data Bank* (PDB). Apresentamos uma versão paralela do algoritmo *Branch & Prune* (BP) para o DDGP. Diferentemente do algoritmo paralelo proposto na literatura para o *Discretizable Molecular Distance Geometry Problem* (DMDGP), este novo algoritmo calcula todas as soluções locais em um sistema de coordenadas comum, de modo que a fase de comunicação, realizada através da utilização do esquema cascata clássico, é bem mais eficiente. Experimentos computacionais mostraram que, em geral, as execuções em que o dobro de processos estão envolvidos demoram cerca de metade do tempo. No entanto, quando o número de processos é grande, a fase de comunicação pode ficar mais cara do que as chamadas locais para os BPs sequenciais, devido ao elevado número de distâncias entre os vértices atribuídos à diferentes processos. Infelizmente, nem todas as instâncias do DDGP podem ser paralelizadas.

Posteriormente, abordamos o DMDGP com distâncias intervalares, denominado como *Interval Discretizable Molecular Distance Geometry Problem* (*i*DMDGP) e a versão intervalar do BP, o *Interval Branch & Prune* (*i*BP). Para satisfazer as hipóteses do *i*DMDGP, é necessário a elaboração de uma ordem artificial dos átomos (conforme, Cap. 5). Neste sentido, propusemos uma nova ordem para a cadeia principal de uma molécula de proteína que satisfaz as hipóteses do *i*DMDGP. Essa ordem (veja, Figura 5.13) envolve um número menor de vértices e explora algumas propriedades químicas da cadeia principal da proteína que não foram utilizadas numa ordem previamente definida na literatura (veja, Figura 5.8). Experimentos computacionais mostraram a eficácia da nova ordem.

Por último, abordamos o *Discretizing Vertex Order Problem* (DVOP) que tem

como objetivo encontrar uma ordem para os vértices de V tal que as hipóteses do DDGP sejam satisfeitas. Assim, esse problema é uma importante etapa de pré-processamento para a solução de DDGPs. Apresentamos um novo algoritmo para o DVOP. Experimentos computacionais foram feitos em um conjunto de instâncias do *Sensor Network Localization Problem* (SNLP). Neste caso, algoritmos para o DVOP encontram uma ordem para o SNLP que satisfaz as condições do DDGP. Os resultados computacionais mostraram que o novo algoritmo proposto para o DVOP tem melhores resultados em relação ao tempo computacional do que um algoritmo proposto anteriormente na literatura. Destaca-se a importância de algoritmos que resolvam o DVOP eficientemente, pois representa uma importante etapa de pré-processamento para a solução de DDGPs.

No sentido de continuação da pesquisa, elencamos a seguir alguns pontos a serem investigados em trabalhos futuros:

- No que diz respeito à paralelização do BP para o DDGP. Verificamos que, nem todas as instâncias do DDGP podem ser paralelizadas. Elas precisam satisfazer algumas suposições, ou seja, os pressupostos na Definição 3.1. Assim, pode-se investigar, para um dado p, como podemos transformar uma instância DDGP em uma instância DDGP p-paralelizável.

- Com relação às ordens dos átomos para a cadeia principal de proteína, pesquisas futuras podem ser destinadas a analisar em mais detalhes e melhorar essa nova ordem proposta. A idéia principal é encontrar ordens que possam melhorar o desempenho do algoritmo iBP.

- No que se refere a algoritmos para o DVOP, pretende-se desenvolver com mais detalhes a teoria associada ao algoritmo proposto para este problema, além de explorar a possibilidade de combinar o algoritmo proposto com algoritmos da literatura.

Referências Bibliográficas

[1] BERMAN, H. M., WESTBROOK, J., FENG, Z., GILLILAND, G., BHAT, T. N., WEISSIG, H., SHINDYALOV, I. N., BOURNE, P. E., 2000, "The Protein Data Bank", *Nucleic Acids Research*, v. 28, pp. 235 – 242.

[2] BISWAS, P., TOH, K.-C., YE, Y., 2008, "A distributed SDP approach for large-scale noisy anchor-free graph reailzation with applications to molecular conformation", *SIAM Journal on Scientific Computing*, v. 30, n. 3, pp. 1251–1277. ISSN: 1064-8275.

[3] BISWAS, P., LIAN, T.-C., WANG, T.-C., YE, Y., 2006, "Semidefinite programming based algorithms for sensor network localization", *ACM Transactions on Sensor Networks*, v. 2, n. 2, pp. 188–220. ISSN: 1550-4859.

[4] BLUMENTHAL, L. M., 1953, *Theory and applications of distance geometry*. Oxford, Clarendon Press.

[5] BRENNER, S. E., 2001, "A tour of structural genomics", *Nat Rev Genet*, v. 2, n. 10 (out.), pp. 801–809. ISSN: 1471-0056.

[6] BRÜNGER, A. T., NILGES, M., 1993, "Computational challenges for macromolecular structure determination by X-ray crystallography and solution NMR-spectroscopy." *Quarterly Reviews of Biophysics*, v. 26, n. 1, pp. 49–125.

[7] CARVALHO, R., LAVOR, C., PROTTI, F., 2008, "Extending the geometric build-up algorithm for the molecular distance geometry problem", *Inf. Process. Lett.*, v. 108, n. 4, pp. 234–237. ISSN: 0020-0190.

[8] CHUNG, F. R. K., GARRETT, M. W., GRAHAM, R. L., SHALLCROSS, D., 2001, "Distance Realization Problems with Applications to Internet Tomography", *Journal of Computer and System Sciences*, v. 63, n. 3, pp. 432–448.

[9] COOPE, I. D., 2000, "Reliable computation of the points of intersection of n spheres in n-space", *(Australian and New Zealand Industrial and Applied Mathematics Journal*, v. 42, pp. 461–477.

[10] COSTA, J. A., PATWARI, N., HERO, III, A. O., 2006, "Distributed weighted-multidimensional scaling for node localization in sensor networks", *ACM Transactions on Sensor Networks*, v. 2 (February), pp. 39–64. ISSN: 1550-4859.

[11] CREIGHTON, T., 1993, *Proteins: Structures and Molecular Properties,* 2nd ed. New York, W.H. Freeman.

[12] CRIPPEN, G., HAVEL, T., 1988, *Distance Geometry and Molecular Conformation.* New York, Research Studies Press Ltd.

[13] DATTORRO, J., 2006, *Convex Optimization & Euclidean Distance Geometry.* Lulu.com. ISBN: 1847280641.

[14] DONG, Q., WU, Z., 2002, "A linear-time algorithm for solving the molecular distance geometry problem with exact inter-atomic distances", *Journal of Global Optimization*, v. 22, n. 1-4, pp. 365–375. ISSN: 0925-5001.

[15] GLEICH, D. F., ZHUKOV, L., RASMUSSEN, M., LANG, K., 2005, "The World of Music: SDP Embedding of High Dimensional data". In: *Information Visualization 2005.* Interactive Poster.

[16] GOLUB, G. H., VAN LOAN, C. F., 1989, *Matrix Computations,* 2nd ed. Baltimore, Johns Hopkins University Press.

[17] GRAMACHO, W., 2008, *Algoritmos para o Cálculo de Estruturas de Proteínas.* Tese de Mestrado, UFF.

[18] GRAMACHO, W., MUCHERINO, A., MACULAN, N., 2012, "A Parallel BP Algorithm for the Discretizable Distance Geometry Problem". In: *Parallel and Distributed Processing Symposium Workshops PhD Forum (IPDPSW), 2012 IEEE 26th International*, pp. 1762–1768.

[19] GRAMACHO, W., GONÇALVES, D., MUCHERINO, A., MACULAN, N., 2013, "A new algorithm to finding discretizable orderings for Distance Geometry". In: *Workshop on Distance Geometry and Applications, 2013*, pp. 149–152.

[20] GROPP, W., LUSK, E., 1996. "User's Guide for mpich, a Portable Implementation of MPI Version 1.2.1". .

[21] HAVEL, T. F., 1996, "Distance Geometry". In: *Encyclopedia of Nuclear Magnetic Resonance*, pp. 1701–1710. J. Wiley & Sons.

[22] HAVEL, T. F., 1991, "An Evaluation of Computational Strategies for Use in the Determination of Protein Structure from Distance Constraints Obtained by Nuclear Magnetic Resonance". In: *Progress in Biophysics and Molecular Biology*, Permagon Press, pp. 43–78, Oxford, England.

[23] HENDRICKSON, B., 1995, "The molecule problem: Exploiting structure in global optimization", *SIAM Journal on Optimization*, v. 5, n. 4, pp. 835–857.

[24] HENDRICKSON, B., 1992, "Conditions for Unique Graph Realizations", *SIAM Journal on Computing*, v. 21, n. 1, pp. 65–84.

[25] KIMMEL, R., 2003, *Numerical Geometry of Images: Theory, Algorithms, and Applications*. Springer Verlag.

[26] KLOCK, H., BUHMANN, J. M., 1997, "Multidimensional Scaling by Deterministic Annealing". In: *EMMCVPR '97: Proceedings of the First International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pp. 245–260, London, UK. Springer-Verlag. ISBN: 3-540-62909-2.

[27] KRISLOCK, N., WOLKOWICZ, H., 2010, "Explicit Sensor Network Localization using Semidefinite Representations and Facial Reductions", *SIAM J. on Optimization*, v. 20, n. 5 (jul.), pp. 2679–2708.

[28] LAVOR, C., 2006, "Global Optimization: from Theory to Implementation". in liberti, l. and maculan, n. ed., cap. On generating instances for the molecular distance geometry problem, Berlin, Springer.

[29] LAVOR, C., LIBERTI, L., MACULAN, N., 2006. "The Discretizable Molecular Distance Geometry Problem". ago.. Disponível em: <http://arxiv.org/abs/q-bio.BM/0608012>.

[30] LAVOR, C., LIBERTI, L., MACULAN, N., 2006, "Global Optimization: Scientific and Engineering Case Studies". in pintér, j. ed., cap. Computational Experience With The Molecular Distance Geometry Problem, New York, Springer, .

[31] LAVOR, C., LIBERTI, L., MACULAN, N., 2008, "An overview of distinct approaches for the molecular distance geometry problem". In: Pardalos, P., Floudas, C. (Eds.), *Encyclopedia of Optimization*, Berlin. 2nd Edition.

[32] LAVOR, C., LIBERTI, L., MACULAN, N., MUCHERINO, A., 2011, "Recent advances on the discretizable molecular distance geometry problem", *European Journal of Operational Research*, (nov.). ISSN: 03772217.

[33] LAVOR, C., LIBERTI, L., MUCHERINO, A., 2011, "The interval Branch-and-Prune algorithm for the discretizable molecular distance geometry problem with inexact distances", *Journal of Global Optimization*, pp. 1–17.

[34] LAVOR, C., LEE, J., JOHN, A. L.-S., LIBERTI, L., MUCHERINO, A., SVIRIDENKO, M., 2012, "Discretization orders for distance geometry problems." *Optimization Letters*, v. 6, n. 4, pp. 783–796.

[35] LAVOR, C., GRAMACHO, W., MUCHERINO, A., LIBERTI, L., 2013, "A new discretization order for protein backbones". In: *9th International Symposium on Bioinformatics Research and Applications (ISBRA13).*

[36] LIBERTI, L., LAVOR, C., MACULAN, N., 2005, "Double VNS for the Molecular Distance Geometry Problem". In: *Proceedings of Mini Euro Conference on Variable Neighbourhood Search*, pp. 23–5, Tenerife, Spain.

[37] LIBERTI, L., LAVOR, C., MACULAN, N., 2008, "A Branch-and-Prune Algorithm for the Molecular Distance Geometry Problem", *International Transactions in Operational Research*, v. 15, n. 1, pp. 1–17.

[38] LIBERTI, L., LAVOR, C., MACULAN, N., 2008, "A Branch-and-Prune algorithm for the Molecular Distance Geometry Problem", *International Transactions in Operational Research*, v. 15, n. 1, pp. 1–17.

[39] LIBERTI, L., LAVOR, C., MUCHERINO, A., MACULAN, N., 2011, "Molecular distance geometry methods: from continuous to discrete", *International Transactions in Operational Research*, v. 18, n. 1, pp. 33–51. ISSN: 1475-3995.

[40] LIBERTI, L., LAVOR, C., MACULAN, N., MUCHERINO, A., 2012. "Euclidean distance geometry and applications". arXiv:1205.0349.

[41] LUSK, E., DOSS, N., SKJELLUM, A., 1996, "A high-performance, portable implementation of the MPI message passing interface standard", *Parallel Computing*, v. 22, pp. 789–828.

[42] MI YOON, J., GAD, Y., WU, Z., 2000, *Mathematical Modeling of Protein Structure Using Distance Geometry.* Relatório Técnico TR0024, Computational and Applied Mathematics Department of Rice University, jul.

[43] MORÉ, J. J., WU, Z., 1996, "ϵ-optimal solutions to distance geometry problems via global continuation". In: *Global minimization of nonconvex energy functions: molecular conformation and protein folding*, American Mathematical Society, pp. 151–168, Providence, RI.

[44] MORÉ, J. J., WU, Z., 1999, "Distance geometry optimization for protein structures", *Journal of Global Optimization*, v. 15, n. 3, pp. 219–234.

[45] MORÉ, J. J., WU, Z., 1997, "Global Continuation for Distance Geometry Problems", *SIAM Journal on Computing*, v. 7, n. 3, pp. 814–836. ISSN: 1052-6234.

[46] MORÉ, J. J., WU, Z., 1996, "Smoothing techniques for macromolecular global optimization". In: Press, P. (Ed.), *Nonlinear Optimization and Applications*, Di Pillo, G. and Giannessi, F., pp. 297–312, New York.

[47] MUCHERINO, A., LAVOR, C., LIBERTI, L., TALBI, E.-G., 2010, "A parallel version of the Branch & Prune algorithm for the Molecular Distance Geometry Problem". In: *Proceedings of the ACS/IEEE International Conference on Computer Systems and Applications - AICCSA 2010*, AICCSA '10, pp. 1–6, Washington, DC, USA. IEEE Computer Society. ISBN: 978-1-4244-7716-6.

[48] MUCHERINO, A., LAVOR, C., LIBERTI, L., 2011, "The discretizable distance geometry problem", *Optimization Letters*, (jun.), pp. 1–16. ISSN: 1862-4472.

[49] NILGES, M., MACIAS, M. J., OSCHKINAT, H., 1997, "Automated NOESY interpretation with ambiguous distance restraints: the refined NMR solution structure of the pleckstrin homology domain from b-spectrin", *J. Mol. Biol*, pp. 408–422.

[50] PATWARI, N., HERO, III, A. O., PACHOLSKI, A., 2005, "Manifold learning visualization of network traffic data". In: *Proceedings of the 2005 ACM SIGCOMM workshop on Mining network data*, MineNet '05, pp. 191–196, New York, NY, USA. ACM. ISBN: 1-59593-026-4.

[51] PHILLIPS, A., ROSEN, J., WALKE, V., 1996. "Molecular structure determination by convex global underestimation of local energy minima". .

[52] SAXE, J. B., 1979, "Embeddability of weighted graphs in k-space is strongly NP-hard", *Proc. 17th Allerton Conference in Communications, Control and Computing*, pp. 480–489.

[53] SCHLICK, T., 2002, *Molecular Modeling and Simulation: An Interdisciplinary Guide*. Secaucus, NJ, USA, Springer-Verlag New York, Inc. ISBN: 038795404X.

[54] SCHLICK, T., 2002, *Molecular Modeling and Simulation: An Interdisciplinary Guide*. Secaucus, NJ, USA, Springer-Verlag New York, Inc. ISBN: 038795404X.

[55] SIPPL, M. J., SCHERAGA, H. A., 1986, "Cayley-Menger coordinates." *Proceedings of the National Academy of Sciences of the United States of America*, v. 83, pp. 2283–2287.

[56] TROSSET, M. W., 1998, "Applications of Multidimensional Scaling to Molecular Conformation", *Computing Science and Statistics*, v. 29, n. 1, pp. 148–152.

[57] VITKUP, D., MELAMUD, E., MOULT, J., SANDER, C., 2001, "Completeness in structural genomics." *Nat Struct Biol*, v. 8, n. 6 (jun.), pp. 559–566.

[58] WEINBERGER, K., SAUL, L., 2006, "Unsupervised Learning of Image Manifolds by Semidefinite Programming", *International Journal of Computer Vision*, v. 70, n. 1 (out.), pp. 77–90. ISSN: 0920-5691.

[59] WU, D., WU, Z., 2007, "An updated geometric build-up algorithm for solving the molecular distance geometry problems with sparse distance data", *Journal of Global Optimization*, v. 37, n. 4, pp. 661–673. ISSN: 0925-5001.

[60] ZOU, Z., BIRD, R. H., SCHNABEL, R. B., 1997, "A stochastic/perturbation global optimization algorithm for distance geometry problems", *Journal of Global Optimization*, v. 11, n. 1, pp. 91–105.