

**UTFPR – UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CAMPUS CORNÉLIO PROCÓPIO
PROGRAMA DE PÓS-GRADUAÇÃO EM BIOINFORMÁTICA**

ANDREY CABRAL MEIRA

**ABORDAGEM BIOBJETIVO PARA O PROBLEMA DE
DOBRAMENTO DE PROTEÍNAS UTILIZANDO O MODELO 3DHPSC**

DISSERTAÇÃO

CORNÉLIO PROCÓPIO

2021

ANDREY CABRAL MEIRA

**ABORDAGEM BIOBJETIVO PARA O PROBLEMA DE
DOBRAMENTO DE PROTEÍNAS UTILIZANDO O MODELO 3DHPSC**
*BIOBJECTIVE APPROACH TO PROTEIN FOLDING PROBLEM USING
3DHPSC MODEL*

Dissertação apresentada ao PROGRAMA DE
PÓS-GRADUAÇÃO EM BIOINFORMÁTICA
da UTFPR- UNIVERSIDADE TECNOLÓGICA
FEDERAL DO PARANÁ CÂMPUS CORNÉLIO
PROCÓPIO como requisito para a obtenção do grau
de “Mestre em Bioinformática”.

Orientador: Prof. Dr. César Manuel Vargas
Benítez

Co-orientador: Prof. Dr. Danilo Sipoli Sanches

CORNÉLIO PROCÓPIO

2021



[4.0 Internacional](https://creativecommons.org/licenses/by-nc-nd/4.0/)

Esta licença permite download e compartilhamento do trabalho desde que sejam atribuídos créditos ao(s) autor(es), sem a possibilidade de alterá-lo ou utilizá-lo para fins comerciais. Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença



Ministério da Educação
Universidade Tecnológica Federal do Paraná
Câmpus Cornélio Procópio



ANDREY CABRAL MEIRA

**ABORDAGEM BIOBJETIVO PARA O PROBLEMA DE DOBRAMENTO DE PROTEÍNAS UTILIZANDO O
MODELO 3DHPSC**

Trabalho de pesquisa de mestrado apresentado como requisito para obtenção do título de Mestre Em Bioinformática da Universidade Tecnológica Federal do Paraná (UTFPR). Área de concentração: Bioinformática.

Data de aprovação: 03 de Dezembro de 2020

Prof Cesar Manuel Vargas Benitez, - Universidade Tecnológica Federal do Paraná

Prof.a Denise Fukumi Tsunoda, Doutorado - Universidade Federal do Paraná (Ufpr)

Prof Fabio Fernandes Da Rocha Vicente, Doutorado - Universidade Tecnológica Federal do Paraná

Documento gerado pelo Sistema Acadêmico da UTFPR a partir dos dados da Ata de Defesa em 03/12/2020.

Este trabalho é dedicado a todos que participam e participaram da minha vida de todas as formas possíveis.

Em especial a minha mãe Hermínia Cabral e minha irmã Andressa Cabral Meira que estiveram ao meu lado nessa trajetória.

Dedico este trabalho também a minha irmã Mariane Meira Oliveira que se foi durante esse período, mas tenho certeza que está torcendo o tempo todo por mim de um lugar melhor.

AGRADECIMENTOS

Agradeço a todos os professores do PPGBIOINFO-CP pelas oportunidades, experiências e debates neste período de mestrado. Em especial aos professores Alessandro Botelho Bovo e André Yoshiaki Kashiwabara pela oportunidade antes de me tornar aluno e me abrirem os olhos para as possibilidades, ao professor Fabrício Lopes Martins pelas ajudas e indicações durante o curso, ao professor Alexandre Rossi Paschoal pelas diversas conversas e parceria, ao professor Danilo Sipoli Sanches por intervir por mim quando precisei e ao professor e orientador César Manoel Vargas Benítez pela oportunidade e pela calma e compreensão durante este tempo, sem isso não conseguiria chegar ao fim.

Aos professores Cristiano Marcos Agulhari, Marcio Mendonça, Adailton Silva Borges, Fabian Andres Lara Molina e Rodrigo Sumar por me deixarem aprimorar meus conhecimentos em suas disciplinas como aluno externo e pelas discussões.

Agradeço aos colegas do Departamento de computação da UTFPR-CP por me acolherem quando entrei como professor substituto, me fornecendo informações e estrutura para poder desenvolver meu trabalho. Em especial aos professores Willian Massami Watanabe, Katia Romero Felizardo Scannavino, Adriana Herden, Erica Ferreira de Souza, Alexandre Rômolo Moreira Feitosa, Antônio Carlos Fernandes da Silva, Eduardo Cotrin Teixeira, Roberto Sadao e Gisele Alves Santana por todo apoio nas disciplinas e dúvidas que tive sobre conteúdos e materiais.

Aos amigos que fiz na pós graduação, Lahyon Roberto, Calaquela João Manoel, Daniel Cordeiro, Daniel Longhi, Robson Parmezan e José Eduardo, dividimos as dificuldades durante o processo e nos apoiamos sempre que possível.

Aos meus amigos de república por terem me acolhido quando cheguei em Cornélio Procópio e a todos os amigos que fiz durante este período. Em especial aos amigos Carlos Franco Thiago Rodrigues e Fernando Zolin.

Aos amigos de longa data Marcos Fernando, Bruno Gabriel, Fernando Rech, Claudio Lucas, Lincoln Keiser, Luiz Gabriel, Marlon Verão, Pedro Henrique, Anderson Maciel, Caio Cezar, Lucas Eduardo, Willian Oliveira e Guilherme Bianchini pela parceria de sempre.

Agradeço aos professores Robson Soares e Marcos Alves e o amigo Bruno por me dar

uma força para entrar no mestrado.

Agradeço a minha mãe e ao meu tio Silvio Meira e sua família, por ter chegado até este ponto e por estarem do meu lado sempre que possível e que foi necessário.

Agradeço Deus pelo dom da vida e pelas oportunidades.

RESUMO

CABRAL, Andrey. ABORDAGEM BIOOBJETIVO PARA O PROBLEMA DE DOBRAMENTO DE PROTEÍNAS UTILIZANDO O MODELO 3DHPSC. 126 f. Dissertação – PROGRAMA DE PÓS-GRADUAÇÃO EM BIOINFORMÁTICA, UTFPR – UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ CAMPUS CORNÉLIO PROCÓPIO . Cornélio Procópio, 2021.

O problema de dobramento de proteínas tem sido vastamente explorado por diversos campos das ciências em busca de uma forma de se obter estruturas funcionais a partir de diversos tipos de modelagens. Os modelos computacionais vem sendo parte fundamental em tais estudos, buscando prever e validar as estruturas geradas.

Este projeto apresenta uma proposta bioinspirada bio-objetivo para abordar o modelo 3DHPSC de dobramento de proteínas, que foi pouco explorado pela literatura. O algoritmo NSGA-II foi utilizado neste estudo em conjunto com sequências *benchmark* para validação do método. Os resultados foram comparados e discutidos com os obtidos pelo método mono objetivo apresentado na literatura.

Palavras-chave: Benchmark, Codificação, PDP, NSGA-II

ABSTRACT

CABRAL, Andrey. BIOBJECTIVE APPROACH TO PROTEIN FOLDING PROBLEM USING 3DHPSC MODEL. 126 f. Dissertação – PROGRAMA DE PÓS-GRADUAÇÃO EM BIOINFORMÁTICA, UTFPR – UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ CAMPUS CORNÉLIO PROCÓPIO . Cornélio Procópio, 2021.

The protein folding problem has been extensively explored by several fields of science in search of a way to obtain functional structures from different types of modeling. Computational models have been a fundamental part of such studies, seeking to predict and validate the structures generated.

This project presents a bioinspired bio-objective proposal to approach the 3DHPSC model of protein folding, which has been little explored in the literature. The NSGA-II algorithm was used in this study in conjunction with textit benchmark sequence to validate the method. The results were compared and discussed with those obtained by the single objective method presented in the literature.

Keywords: Benchmark, Codification, PFP, NSGA-II

LISTA DE FIGURAS

FIGURA 1	– Representação de um aminoácido	19
FIGURA 2	– Representação dos 20 aminoácidos naturais	21
FIGURA 3	– Representação de uma ligação peptídica entre 2 aminoácidos	22
FIGURA 4	– Representação dos ângulos de torção	23
FIGURA 5	– Estruturas Secundárias	24
FIGURA 6	– Estruturas de proteínas	26
FIGURA 7	– Conjuntos de movimentos em Lattice 2D	27
FIGURA 8	– Conjuntos de movimentos em Lattice 3D	28
FIGURA 9	– Exemplo de conformação no modelo 3DHPS	28
FIGURA 10	– Modelo FCC	29
FIGURA 11	– Conformação pelo modelo FCC	30
FIGURA 12	– Modelo UNRES	36
FIGURA 13	– Astro-Fold	38
FIGURA 14	– Exemplo da distribuição nos tipos de seleção do algoritmo genético	45
FIGURA 15	– Exemplos de conjuntos de Pareto-Ótimo	49
FIGURA 16	– Exemplo de ordenação das Fronteiras por dominância	53
FIGURA 17	– Distância de multidão de um ponto	56
FIGURA 18	– Panorama geral do NSGA-II	57
FIGURA 19	– Exemplo da codificação e decodificação dos indivíduos	64
FIGURA 20	– Primeiro backbone com sua respectiva cadeia lateral	65
FIGURA 21	– Orientação da primeira estrutura	66
FIGURA 22	– Orientação do segundo <i>backbone</i>	66
FIGURA 23	– Orientação da segunda Cadeia Lateral relativa ao segundo <i>backbone</i>	67
FIGURA 24	– Orientação do Terceiro <i>backbone</i>	68
FIGURA 25	– Orientação da terceira Cadeia Lateral relativa ao terceiro <i>backbone</i>	69
FIGURA 26	– Cruzamento de dois pontos	70
FIGURA 27	– Exemplo da alteração do bit mais significativo do vetor de movimentos ..	77
FIGURA 28	– Estruturas S48.5 com o mesmo número de contatos hidrofóbicos, mas com energia livre diferentes	82
FIGURA 29	– HNC X F2 S48.5	85
FIGURA 30	– HNC S48.5	86
FIGURA 31	– F2 S48.5	87
FIGURA 32	– Estruturas S48.6 , S48.7 e S48.8	89
FIGURA 33	– Estrutura e Funções para Dill.1	102
FIGURA 34	– Estrutura e Funções para Dill.2	103
FIGURA 35	– Estrutura e Funções para Dill.3	104

FIGURA 36	– Estrutura e Funções para Dill.4	105
FIGURA 37	– Estrutura e Funções para Dill.5	106
FIGURA 38	– Estrutura e Funções para Unger273d.1	107
FIGURA 39	– Estrutura e Funções para Unger273d.2	108
FIGURA 40	– Estrutura e Funções para Unger273d.3	109
FIGURA 41	– Estrutura e Funções para Unger273d.4	110
FIGURA 42	– Estrutura e Funções para Unger273d.5	111
FIGURA 43	– Estrutura e Funções para Unger273d.6	112
FIGURA 44	– Estrutura e Funções para Unger273d.7	113
FIGURA 45	– Estrutura e Funções para Unger273d.8	114
FIGURA 46	– Estrutura e Funções para Unger273d.9	115
FIGURA 47	– Estrutura e Funções para Unger273d.10	116
FIGURA 48	– Estrutura e Funções para S48.1	117
FIGURA 49	– Estrutura e Funções para S48.2	118
FIGURA 50	– Estrutura e Funções para S48.3	119
FIGURA 51	– Estrutura e Funções para S48.4	120
FIGURA 52	– Estrutura e Funções para S48.5	121
FIGURA 53	– Estrutura e Funções para S48.6	122
FIGURA 54	– Estrutura e Funções para S48.7	123
FIGURA 55	– Estrutura e Funções para S48.8	124
FIGURA 56	– Estrutura e Funções para S48.9	125
FIGURA 57	– Estrutura e Funções para S48.10	126

LISTA DE TABELAS

TABELA 1	– Lista dos principais Algoritmos	33
TABELA 2	– Movimentos relativos BB-SC	64
TABELA 3	– Pesos para as interações do sistema utilizado (Benitez e Lopes 2010)	71
TABELA 4	– Sequências de <i>Benchmark</i> para o modelo 3DHPSC	75
TABELA 5	– Resultados para o <i>dataset</i> Dill.*	78
TABELA 6	– Resultados para o <i>dataset</i> Unger273d.*	79
TABELA 7	– Resultados para o <i>dataset</i> S48.*	80
TABELA 8	– Contatos por experimento no caso S48.5	83
TABELA 9	– Pesos para as interações do sistema utilizado (Benitez e Lopes 2010)	83

LISTA DE ABREVIATURAS E SIGLAS

α BB	Alfa branch And Bound
2D	Bidimensional
3D	Tridimensional
ABC	Artificial Bee Colony
ACO	Ant Colony Optimization
AGs	Algoritmos Genéticos
AMBER	Assisted Model Building with Energy Refinement
BA	Bat Algorithm
CAMEO	Continuous Automated Model EvaluatiO
CASP	Critical Assessment of protein Structure Prediction
CE	Computação Evolucionnária
CHARMM	Chemistry at HARvard Molecular Mechanics
CSA	Conformational Space Annealing
EMC	Evolutionary Monte Carlo
FCC	Face- Centred cubic
GWO	Grey Wolf Optimizer
HP	Hidrofóbico Polar
IA	Inteligência Artificial
IE	Inteligência de Enxames
KNN	K-Nearest-Neighbours
LSTM	Long Short-Term Memory
MC	Monte Carlo
MD	Dinâmica Molecular
MOEA	Multi-Objective Evolutionary Algorithm

MOEA/D	MultiObjective Evolutionary Algorithm Based on Decomposition
MOGA	Multiple Objective Genetic Algorithm
MOOP	Multi-Objective Optimization Problem
NP-Difícil	Não Polinomial Difícil
NPGA	Niched-Pareto Genetic Algorithm
NSGA	Non-dominated Sorting Genetic Algorithm
NSGA-II	Non-dominated Sorting Genetic Algorithm II
OMETS	Local Meta-Threading-Server
OPLS	Optimized Potentials for Liquid Simulations
PAES	Pareto Archived Evolution Strategy
PDB	Protein Data Bank
PDP	Problema de Predição de proteínas
PSO	Particle Swarming Optimization
REMC	Replica exchange Monte Carlo
SA	Simulated Annealing
SC	Side Chain
SPEA	Strength Pareto Evolutionary Algorithm
TrEMBL	Translated EMBL Nucleotide Sequence Data Library
UniProtKB	The Universal Protein Resource
UNRES	United Residue
VEGA	Vector Evaluated Algorithm
VOES	VecVector Optimized Evolution Strategy
WBGA	Weighted Based Genetic Algorithm

LISTA DE SÍMBOLOS

$C\alpha$	Carbono Alfa
NH_3^-	Amina
$COOH$	Carboxila
H_2O	Água
R	Radical - Resíduo
OH^-	Hidroxil
ϕ	Ângulo da ligação entre o N e o $C\alpha$ (Phi)
ψ	Ligação entre o $C\alpha$ e o C da carboxila(Psi)
ω	Ângulo entre C da carboxila e o N da amina (Ômega)
H	Hidrofóbico
P	Polar

SUMÁRIO

1 INTRODUÇÃO	15
1.1 OBJETIVOS	17
1.1.1 Objetivo Geral	17
1.1.2 Objetivos Específicos	17
1.1.3 Organização do Trabalho	17
2 FUNDAMENTAÇÃO TEÓRICA	18
2.1 PROTEÍNAS, AMINOÁCIDOS E PEPTÍDEOS	18
2.2 ESTRUTURAS DE PROTEÍNAS	22
2.3 PREDIÇÃO DE ESTRUTURA PROTEÍNAS	26
2.3.1 Modelos Baseados em Lattice	27
2.3.1.1 Modelo 3DHPSC	30
2.3.2 Modelos Off-Lattice	33
2.4 COMPUTAÇÃO BIOINSPIRADA	40
2.4.1 Algoritmos Genéticos	43
2.5 ALGORITMOS EVOLUTIVOS MULTIOBJETIVO	47
2.5.1 NSGA-II	52
2.6 TRABALHOS RELACIONADOS	59
3 METODOLOGIA	62
3.1 PROPOSTA BIOBJETIVO PARA O MODELO 3DHPSC	62
4 EXPERIMENTOS E DISCUSSÃO	76
4.1 PARÂMETROS	76
4.2 DISCUSSÃO DOS RESULTADOS	78
5 CONSIDERAÇÕES FINAIS	90
REFERÊNCIAS	93
Apêndice A – ESTRUTURAS GERADAS	101

1 INTRODUÇÃO

As proteínas são essenciais para os seres vivos exercendo funções biológicas importantes no organismo como formação de estruturas (conferindo rigidez por proteínas fibrosas como o colágeno e a queratina), defesa (por exemplo os linfócitos, eosinófilos e basófilos), catalisadoras de reações químicas (como as enzimas), transporte (hemoglobina e ferritina), comunicação (insulina transmitindo sinais para tecidos distantes por exemplo) dentre outros. Além disto, a maior parte da informação genética dos seres vivos é expressa pelas proteínas, sendo todas constituídas por cadeias aminoácidos conectados por ligações peptídicas (PIERCE 2016, Lesk 2014).

Durante a síntese, as proteínas passam por um processo chamado de dobramento, onde elas se contorcem até chegarem em uma estrutura compacta e funcional, conhecida como estrutura nativa. A malformação de proteínas é responsável por doenças como Mal de Alzheimer, fibrose cística, doenças neuronais, alguns tipos de câncer, dentre outras patologias (Soto 2001, Cohen e Kelly 2003, Gregersen et al. 2006, Soto e Estrada 2008). Logo, um melhor entendimento do processo de dobras de proteína pode auxiliar no entendimento dos processos patológicos (Cohen e Kelly 2003, Soto e Estrada 2008) e no desenvolvimento de fármacos para o tratamento (Nunes et al. 2016, Schmidt et al. 2014, Ozboyaci et al. 2016).

O UniProtKB/TrEMBL ¹ contém atualmente cerca de 65 milhões de sequências registradas (Consortium 2016), porém apenas uma pequena parte dessas sequências tem sua estrutura tridimensional conhecida, o Protein Data Bank (PDB)² registra atualmente cerca de 171 mil estruturas (RCBS 2020).

A baixa correlação entre o UniProtKB/TrEMBL e o PDB se deve dificuldade envolvida no entendimento do dobramento de proteínas, tanto do ponto de vista bioquímico quanto computacional e pelo fato dos métodos empregados para obtenção da estrutura tridimensional das proteínas, como a cristalografia, fotografia por raio-x e espectroscopia de ressonância magnética nuclear serem altamente custosos (Dill 1999).

¹Repositório de sequências proteicas

²Repositório de estruturas tridimensionais de proteínas, ácidos nucleicos e estruturas moleculares complexas

Modelagens em nível atômico clássico vem sendo desenvolvidas com o passar dos anos, mas suas aplicações práticas ainda são limitadas por sua eficiência algorítmica e pelo alto custo computacional (Vendruscolo e Dobson 2011, Piana et al. 2014). Por consequência surgiram modelos simplificados do dobramento de proteínas, porém até a mais simples modelagem realizada atualmente é um problema NP-Difícil computacionalmente (Huang et al. 2010, Benítez 2010), o que indica que provavelmente possa não haver um algoritmo que resolva os dobramentos em tempo polinomial (Galvao et al. 2012).

Métodos Bioinspirados estão sendo aplicados nas mais diversas áreas do conhecimento, pois conseguem resultados para problemas de difícil solução ou não lineares com bom tempo computacional. Dentre os métodos Bioinspirados destacam-se os Algoritmos Genéticos (AGs) (Holland 1975), que partem do princípio da evolução natural buscando fornecer soluções baseando-se em um esquema de população de indivíduos que representam pontos de busca no espaço amostral do sistema em analisado.

Dos modelos encontrados na literatura, o modelo 3DHPSC (Tridimensional Hidrofóbico-Polar com Cadeia Lateral (*Side Chain*)) estudado por (Benítez 2010) apresenta melhoras em relação aos modelos 2DHP (Bidimensional Hidrofóbico-Polar) e 3DHP (Tridimensional Hidrofóbico-Polar) vastamente explorados na literatura, pois possui mais características a serem consideradas durante o processo de dobramento possibilitando uma gama maior de possibilidades de representação espacial da estrutura tridimensional da proteína.

O estudo de (Benítez 2010) utilizou algoritmos genéticos em busca de soluções para o dobramento das sequências de aminoácidos estudados, utilizando apenas uma função objetivo para avaliar a qualidade das soluções. Recentemente (Hattori et al. 2020) obteve resultados importantes no estudo do modelo 3DHPSC utilizando o método de programação inteira com sequências reais obtidas no PDB.

Pelo nosso conhecimento, nenhuma abordagem multiobjetivo foi proposta para o modelo 3DHPSC até o momento, logo, uma nova abordagem é proposta neste trabalho em busca de melhores soluções para os *datasets* utilizados no trabalho de (Benítez 2010). A função objetivo do trabalho de (Benítez 2010) foi decomposta em 2, tornando-o um problema biobjetivo a ser estudado. O algoritmo NSGA-II foi utilizado para buscar soluções de acordo com as novas funções objetivo.

1.1 OBJETIVOS

1.1.1 OBJETIVO GERAL

O presente trabalho tem por objetivo geral a utilização do algoritmo NSGA-II para determinar a estrutura de proteínas utilizando o modelo 3DHPSC.

1.1.2 OBJETIVOS ESPECÍFICOS

Como objetivos específicos:

- Implementar o algoritmo *Non-dominated Sorting Genetic Algorithm II* (NSGA-II) (Deb et al. 2000) para o problema de predição de estrutura de proteínas utilizando o modelo 3DHPSC (Benítez 2010);
- Realizar teste com as sequências *benchmark* utilizadas no trabalho de (Benítez 2010);
- Realizar um comparativo dos resultados do algoritmo proposto com os resultados obtidos pelo 3DHPSC mono-objetivo de (Benítez 2010).

1.1.3 ORGANIZAÇÃO DO TRABALHO

Esta dissertação está organizada da seguinte forma:

- No capítulo 2 são apresentados conceitos básicos de proteínas, suas estruturas e suas composições. São apresentados modelos baseados em *lattice* e *off-lattice* e alguns métodos de busca importantes na área. Por fim será apresentado o modelo 3DHPSC que foi utilizado neste trabalho, as métricas de comparação utilizadas para validar os resultados, computação bioinspirada e o algoritmo, foco deste estudo, o *Non-dominated Sorting Genetic Algorithm II* (NSGA-II);
- No capítulo 3, é apresentada a proposta deste trabalho bem como os algoritmos desenvolvidos, sequências utilizadas e as funções utilizadas.
- No capítulo 4 são apresentados e discutidos os resultados obtidos com as sequências *benchmark*.
- No capítulo 5 as considerações finais são apresentadas e novas sugestões para pesquisas futuras são descritas.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 PROTEÍNAS, AMINOÁCIDOS E PEPTÍDEOS

As proteínas sejam das mais simples as mais complexas são formadas por sequências de aminoácidos unidos por ligações peptídicas, em diversas configurações e assumindo diferentes funções no metabolismo. Essas configurações exercem papel fundamental no processo de formação e manutenção do metabolismo, como formação de enzimas, hormônios, anticorpos, fibras, estruturas dentre outras (Nelson e Cox 2011). Para entendimento de proteínas e suas estruturas se faz necessária uma explanação sobre aminoácidos e peptídeos.

Cada aminoácido é formado por um carbono central ($C\alpha$ - Carbono Alfa) preso a um grupo amina (NH_3^-), um grupo carboxila ($COOH$) e um grupo radical (R), sendo a composição geral dos aminoácidos dada por carbono, hidrogênio, nitrogênio e oxigênio. O que define as características bioquímicas é o grupo R, também comumente descrito na literatura como cadeia lateral ou resíduo (PIERCE 2016). A figura 1 representa a estrutura geral de um aminoácido.

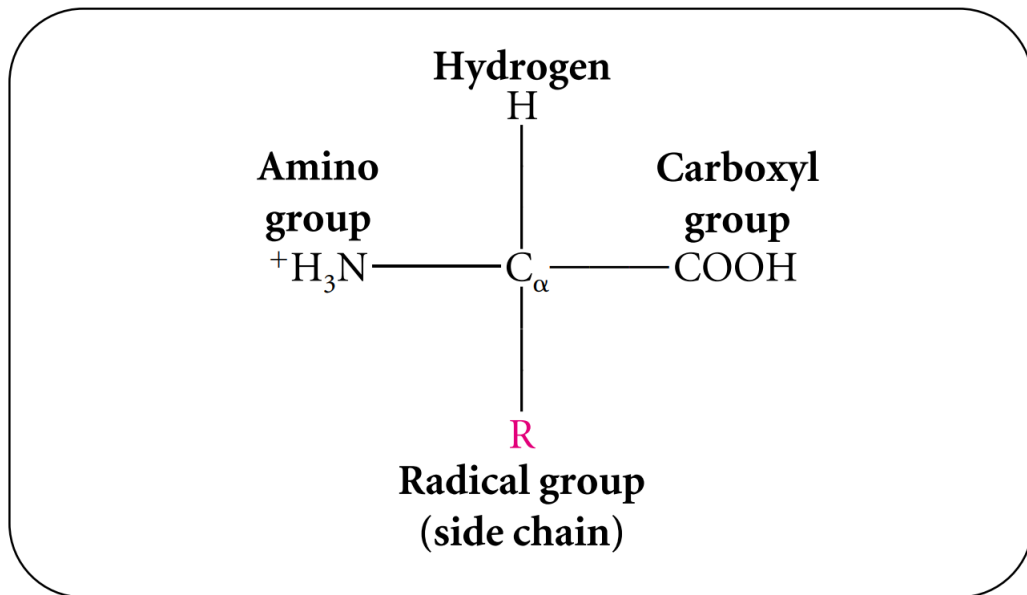


Figura 1: Representação de um aminoácido

Fonte: PIERCE(2016) p.417.

O termo resíduo é empregado devido às ligações covalentes realizadas entre os aminoácidos, denominadas ligações peptídicas, onde uma reação de condensação ocorre e uma molécula de água (H_2O) é liberada. Sendo assim o termo refere-se a perda desta molécula de água.

O grupo R determina a formação de 20 tipos aminoácidos naturais que possuem características diferentes, como estrutura, interação com o solvente, peso molecular, dentre outros.

Esses aminoácidos podem ser agrupados em cinco classes principais com base em suas polaridades ou tendências em interagir com o solvente biológico (água em pH próximo à 7):

Grupos R apolares, alifáticos: São uma classe de aminoácidos polares e hidrofóbicos. Alanina, Valina, Leucina e Isoleucina possuem cadeias laterais que tendem a aglomerar-se entre si nas proteínas, estabilizado-se por interações hidrofóbicas a estrutura proteica.

A Glicina, sendo a estrutura mais simples, é facilmente agrupada com os aminoácidos polares, mas não contribui significativamente para as interações hidrofóbicas. A Metionina, contém um grupo tioéter¹ apolar em sua cadeia lateral.

Grupos R aromáticos: Composto por Fenilalanina, Tirosina e Triptofano. São aminoácidos

¹O tioéter é composto por um átomo de enxofre ligado a dois radicais orgânicos. Um tioéter é um composto orgânico classificado como sulfurado pelo fato de apresentar átomos de enxofre na sua composição.

que possuem cadeias laterais aromáticas relativamente apolares, ou seja hidrofóbicos. O grupo hidroxil (OH^-)² da tirosina pode criar ligações de hidrogênio, sendo um importante grupo funcional em algumas enzimas.

Tirosina e triptofano são mais apolares que a fenilalanina em decorrência do grupo hidroxil da tirosina e do átomo de nitrogênio do anel indólico³ do triptofano.

Grupos R polares, não carregados: Composto por Treonina, Cisteína, Asparagina, Glutamina, Sarina e Prolina. Os resíduos desses aminoácidos são mais solúveis em água, ou mais hidrofílicos se comparados aos aminoácidos apolares, pois contêm grupos funcionais capazes de formar ligações água com hidrogênio. A polaridade da Sarina e Treonina é determinada pelo grupo hidroxil, a Cisteína pelo grupo Sulfidril, um ácido fraco que pode fazer ligações de hidrogênio fracas com átomos de oxigênio e nitrogênio e, da Asparagina e Glutamina por seu grupo amida.

A Prolina possui uma cadeia alifática⁴ com uma estrutura cíclica distinta, o grupo amino secundário dos resíduos de prolina possui uma conformação rígida que reduz flexibilidade estrutural de regiões peptídicas que possuem sua estrutura.

Grupos R carregados positivamente (básicos): Consiste no grupo R mais carregado positivamente quanto negativamente, ou seja mais hidrofílico. Os aminoácidos que possuem carga positiva significativa em pH 7 são Lisina e Arginina. A Histidina possui um grupo aromático imidazol⁵.

O único aminoácido que possui uma cadeia ionizável é a Histidina, que pode estar tanto carregada positivamente quanto não carregada em pH 7. Resíduos de Histidina facilitam diversas reações catalisadas por enzimas ao servirem como aceptores/doadores de prótons.

Grupos carregados negativamente (ácidos): Os dois aminoácidos que compõem este grupo são Aspartato e Glutamato em pH 7. Cada um dos quais possui um segundo grupo carboxil.

Além destes 20 aminoácidos existem outros aminoácidos que não serão abordados pois são estruturas que ocorrem decorrentes de modificações de resíduos já existentes, conhecidos

²Formado por um átomo de hidrogênio e um de oxigênio.

³é um composto orgânico aromático heterocíclico. Possui uma estrutura bicíclica, que consiste em um anel benzênico (6 carbonos) acoplado a um anel de pirrol (anel de 5 membros com um nitrogênio). (Oliveira e Koike 2003)

⁴Em química orgânica, os compostos orgânicos constituídos de carbono e hidrogênio são divididos em duas classes, os aromáticos que contêm anéis benzênicos ou anéis de átomos similares e os alifáticos que são cadeias abertas ou acíclicas.

⁵é um composto orgânico de fórmula $\text{C}_3\text{H}_4\text{N}_2$.

por aminoácidos incomuns. Dentre estes aminoácidos pode-se citar a 4-hidroxiprolina derivada da Prolina, 5-hidroxilisina derivada da Lisina, Selenocisteína derivada da Cisteína, dentre outros. Mais de 300 aminoácidos adicionais foram encontrados em células, eles exercem diversas funções mas nem todos são constituintes de proteínas (Nelson e Cox 2011).

A figura 2 apresenta os 20 aminoácidos naturais em sua representação em bastão, suas abreviações de três letras e os símbolos de uma letra, os quais são utilizados como um atalho para indicar a composição e a sequência dos aminoácidos (Lesk 2014).

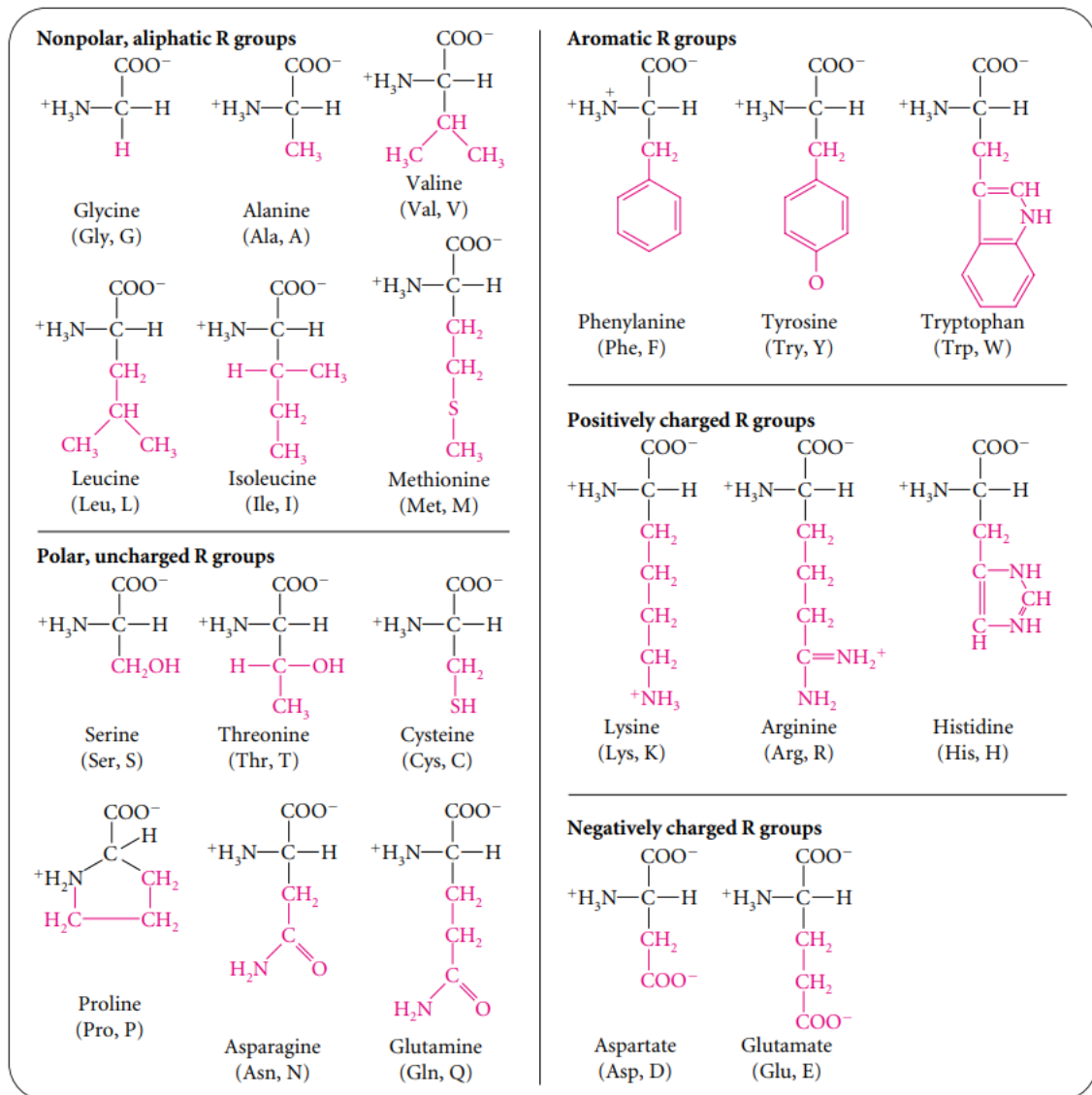


Figura 2: Representação dos 20 aminoácidos naturais

Fonte: Adaptado de PIERCE(2016) p.417.

Os peptídeos são polímeros ⁶ de aminoácidos, onde duas ou mais moléculas de aminoácidos podem ser ligadas covalentemente, formando uma conexão denominada ligação peptídica (Zhang e Cech 1997).

As cadeias polipeptídicas de proteínas possuem uma cadeia principal de estrutura constante e cadeias laterais de sequência variável, ou seja, as estruturas dos aminoácidos são mantidas, exceto em suas conexões onde são perdidas duas moléculas de hidrogênio e uma molécula de oxigênio formando uma molécula de água (H₂O) (Lesk 2014).

A formação da ligação peptídica é um tipo de reação de condensação, uma classe de reações celulares vivas. A figura 3 ilustra uma ligação peptídica entre 2 aminoácidos.

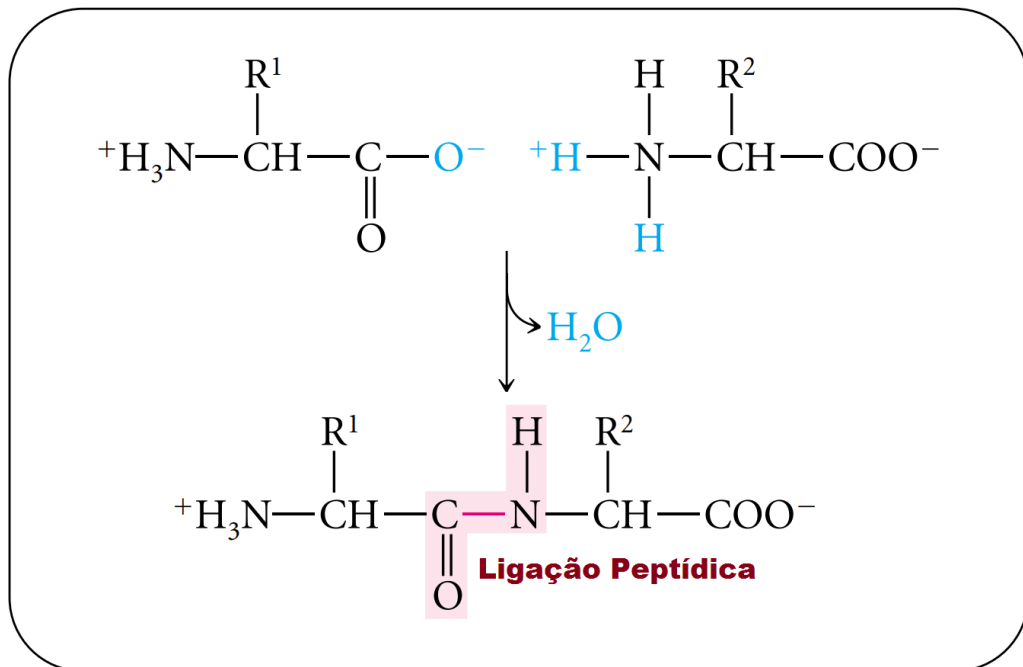


Figura 3: Representação de uma ligação peptídica entre 2 aminoácidos

Fonte: PIERCE(2016) p.418.

Na seção a seguir serão abordados os tipos de estruturas de proteínas e seus níveis organizacionais.

2.2 ESTRUTURAS DE PROTEÍNAS

A ligação peptídica entre dois aminoácidos forma um dipeptídeo, entre três forma um tripeptídeo, entre quatro forma um tetrapeptídeo e assim por diante. A repetição dos grupos amina, C α e carboxílico são denominados pela literatura como cadeia principal (*backbone*).

⁶Polímeros são macromoléculas formadas a partir de unidades estruturais menores (os monômeros).

As conformações das ligações peptídicas são dispostas de 3 ângulos de torção sendo ϕ (phi) a representação do ângulo da ligação entre o N e o $C\alpha$, ψ (psi) a ligação entre o $C\alpha$ e o C da carboxila e ω (ômega) representando o C da carboxila e o N da amina (Lesk 2014). A figura 4 representa os ângulos torção em um *backbone* (área em amarelo).

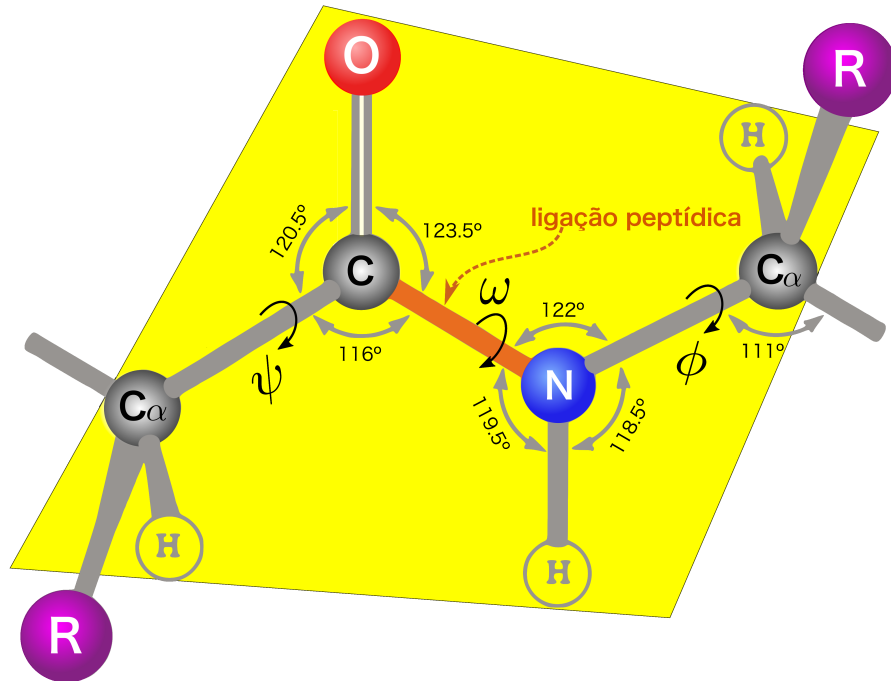


Figura 4: Representação dos ângulos de torção

Fonte: Adaptado de (Almeida et al. 2016) p.35.

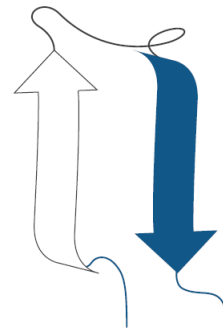
As proteínas são classificadas em quatro níveis organizacionais no que tange ao seu nível de complexidade. A **estrutura primária representa** a sequência linear de aminoácidos, visando apenas representar a sequência da composição proteica.

A **estrutura secundária** representa a primeira fase de conformação das proteínas, representando as primeiras estruturas espaciais desenvolvidas pelas interações moleculares, sendo as α -hélices e as β -folhas as formas mais comuns de estruturas secundárias e, uma estrutura de dobra interna β -turn que se apresenta como uma estrutura de redirecionamento estrutural (Lesk 2014).

Na figura 5 (a) e (b) estão representadas as estruturas de α -hélices e β -folhas.



(a) Representação das α -hélices.



(b) Representação das β -folhas.

Figura 5: Estruturas Secundárias

Fonte: Adaptado de (Almeida et al. 2016) p.40

Muitas proteínas possuem regiões compactas que obedecem um padrão de enovelamento de uma única cadeia, que parece possuir estabilidade independente, denominadas domínios. Domínios são definidos como sendo parte de uma cadeia polipeptídica que é estável e que pode se mover independentemente do restante da proteína. Há também estruturas supersecundárias que são estruturas que apresentam um padrão comum de interações entre α -hlices e β -folhas (Lesk 2014).

Na hierarquia, os domínios ficam entre as estruturas supersecundárias e a estrutura terciária de um monômero integral (Lodish et al. 2014). Polipeptídeos com um pouco mais de algumas centenas de resíduos de aminoácidos frequentemente dobram-se em dois ou mais domínios, podendo até mesmo desempenhar funções biológicas distintas, como a de se ligar a pequenas moléculas ou mesmo a interagir com outras proteínas. Há também as proteínas modulares que são proteínas multidomínio que compreendem a estruturas com muitas cópias do mesmo domínio.

A **estrutura terciária** é o arranjo tridimensional de uma cadeia peptídica, geralmente resultante de diversas interações até uma estrutura com energia livre mínima. Ela resulta de diversas maneiras e não necessariamente precisa passar pelos estados secundários para chegar ao seu estado de energia mínima (ou estrutura nativa) (Weiner et al. 1984). Esta estrutura é estabilizada principalmente pelos efeitos hidrofóbicos, as ligações de hidrogênio entre as cadeias polares e as forças de Van Der Waals⁷ (Alberts et al.).

É importante salientar que estruturas terciárias geralmente são estruturas funcionais no metabolismo, ou seja, não necessariamente precisam interagir com outras proteínas para exercer seu papel metabólico. Muitas proteínas podem ser desnaturadas devido a alterações no meio, como mudanças de temperatura e pH, e voltar a ser funcionais caso atinjam as condições ideais novamente, ou seja, retornarem a sua conformação nativa.

A **estrutura quaternária** é a associação de duas ou mais estruturas terciárias formando macromoléculas. Tendem a formar as estruturas enzimáticas mais complexas dos seres vivos e não são tão instáveis como as estruturas terciárias, devido a sua formação com afinidade molecular com outras estruturas (Lesk 2014).

Proteínas podem ser classificadas por fibrosas e globulares sendo:

- As **proteínas globulares** são cadeias polipeptídicas que atingem forma globular ou esférica. Normalmente contém vários tipos de estruturas secundárias, resultando em

⁷Soma de todas as forças atrativas ou repulsivas, que não sejam forças devidas a ligações covalentes entre moléculas (ou entre partes da mesma molécula) ou forças devido à interação eletrostática de íons

diversas enzimas e proteínas regulatórias.

- As **proteínas fibrosas** são arranjos em forma de fitas, geralmente compostas por estruturas secundárias de β -folhas fortemente ligadas, um exemplo destas estruturas é o colágeno que é responsável pela sustentação da pele (Poian et al. 2010).

A figura 6 apresenta os quatro tipos de estruturas de proteínas.

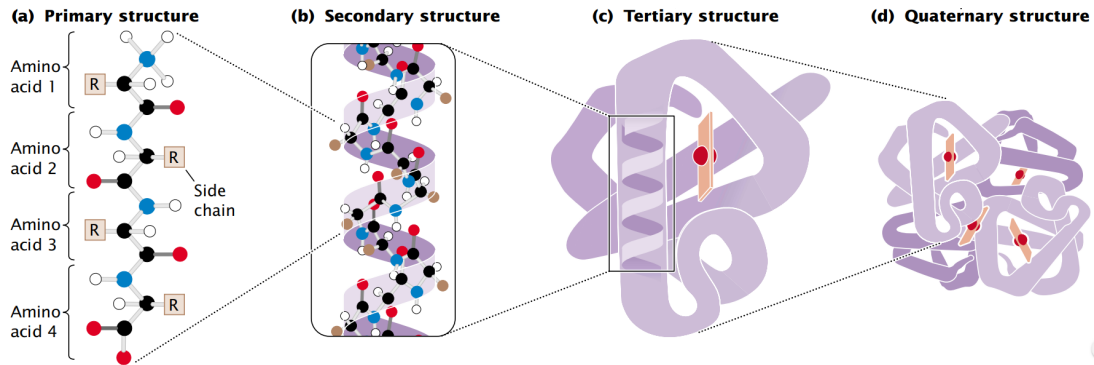


Figura 6: Estruturas de proteínas

Fonte: Adaptado de PIERCE(2016) p.418.

Conhecendo-se os níveis organizacionais das estruturas de proteínas, as próximas seções irão apresentar os modelos computacionais utilizados atualmente no estudo do PDP.

2.3 PREDIÇÃO DE ESTRUTURA PROTEÍNAS

Para estudar o PDP e poder prever estruturas de proteínas, diversos modelos foram desenvolvidos com o passar dos anos. Grande parte destes modelos possuem simplificações e geralmente utilizam as interações moleculares e a energia livre como base para medir a efetividade dos resultados.

Considerando a dificuldade do problema de predição de proteínas (PDP) os modelos podem ser divididos em modelos baseados em *lattices* (reticulados), *off-lattices* (livres) e simulações de todas as estruturas atômicas.

Os modelos em *lattice* utilizam valores discretos como parâmetros tornando sua complexidade menor e por consequência, os resultados menos realistas. Em contra partida possuem maior viabilidade computacional se comparados aos outros modelos, além de conseguir representar alguns comportamentos dos dobramentos, como as interações entre os aminoácidos hidrofóbicos e hidrofílicos (ou polares).

Os modelos *off-lattice* e de simulações atômicas permitem valores contínuos como parâmetros e tendem a apresentar resultados próximos aos reais dependendo do tamanho das proteínas e da complexidade dos campos de força empregados como modelo. Alguns modelos conseguiram simular com boa precisão as estruturas de algumas pequenas proteínas, mas possuem restrições com relação ao desempenho computacional e os tipos de dobramentos representados nas proteínas.

2.3.1 MODELOS BASEADOS EM LATTICE

Nos modelos baseados em *lattice* existe uma limitação com relação a liberdade de movimentos e a representação geralmente se dá em eixos predeterminados sobre o plano (bidimensional ou tridimensional). Estas representações podem ser em treliças por exemplo, utilizadas no modelo Hidrofóbico-Polar (HP) (Dill et al. 1995), onde os aminoácidos são tratados como vértices e as ligações entre eles são arestas de tamanho discreto.

Este modelo se baseia apenas nas interações hidrofóbicas (H) e hidrofílicas (P - polares), sendo que este tipo de modelagem assume que a maior contribuição para a energia livre de uma proteína são as interações entre aminoácidos hidrofóbicos, já que estes aminoácidos tendem a ser compactados no interior das proteínas e os hidrofílicos tendem a ficar na região externa na proteína.

A figura 7 apresenta um conjunto de movimentos simples e uma implementação 2D do modelo HP, onde os vértices representam os aminoácidos (hidrofóbicos ou polares) e as arestas representam as conexões entre eles.

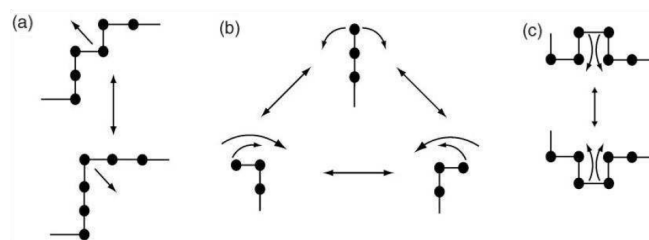


Figura 7: Conjuntos de movimentos em Lattice 2D

Fonte: Retirado de (Venske 2014) p.37.

A figura 8 apresenta uma conformação com as mesmas características da figura 7, porém em um plano tridimensional. Este modelo apresenta maior complexidade com relação ao modelo 2DHP pois apresenta um aumento de possibilidades de contatos entre as cadeias laterais.

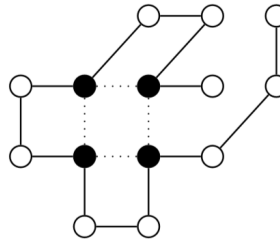


Figura 8: Conjuntos de movimentos em Lattice 3D

Fonte: Retirado de (Benítez 2010) p.42.

Um modelo mais sofisticado do modelo HP, o modelo 3D Hidrofóbico - Polar com cadeia lateral (*Side Chain* - SC) (Li et al. 2002) foi analisado mais recentemente e obteve resultados biologicamente coerentes (Benítez 2010). Neste modelo além dos resíduos (Cadeias Laterais) sendo considerados, as interações com o *backbone*⁸ são consideradas no processo de dobramento. A figura 9 apresenta um dos resultados obtidos nesta modelagem, sendo as esferas azuis representando resíduos hidrofóbicos, os azuis os resíduos hidrofílicos e os *backbones* são representados em cinza.

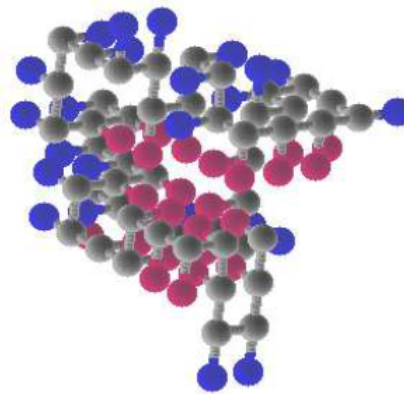


Figura 9: Exemplo de conformação no modelo 3DHPSC

Fonte: Adaptado de (Benítez 2010) p.130.

Uma variação das abordagens em *lattice* foi proposta recentemente baseada em modelagem de face cúbica centrada (FCC - Face-Centred Cubic Model) (Varela e Santos 2017) onde não só os vértices e arestas são consideradas para movimentos, a adição de uma abordagem onde as faces do movimento em cubo são adicionadas no plano 3D (Figura 10).

⁸Nome que se dá aos átomos N-C α -C encadeados

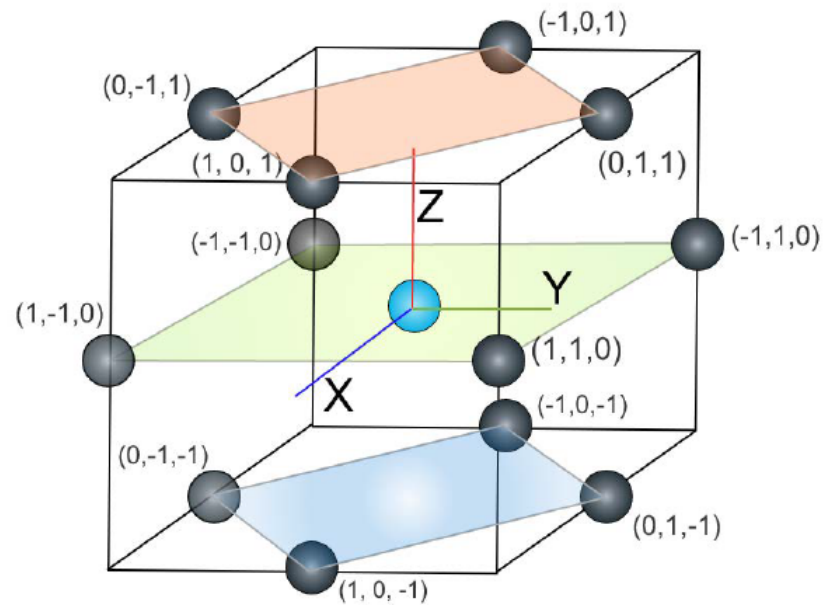


Figura 10: Modelo FCC

Fonte: Retirado de (Varela e Santos 2017) p.2.

Este modelo pela maior liberdade de movimentos para as conexões entre os aminoácidos obteve resultados promissores (Figura 11), pois consegue representar as características do dobramento de acordo com a afinidade do aminoácido com o solvente. Em branco estão representadas os aminoácidos hidrofóbicos e em azul os aminoácidos polares. Devido a simplicidade dos modelos em *lattice* eles são estudados em diversos detalhes, tanto computacionais quanto biológicos, permitindo a compreensão de diversos mecanismos físicos do dobramento de proteínas (Venske 2014).

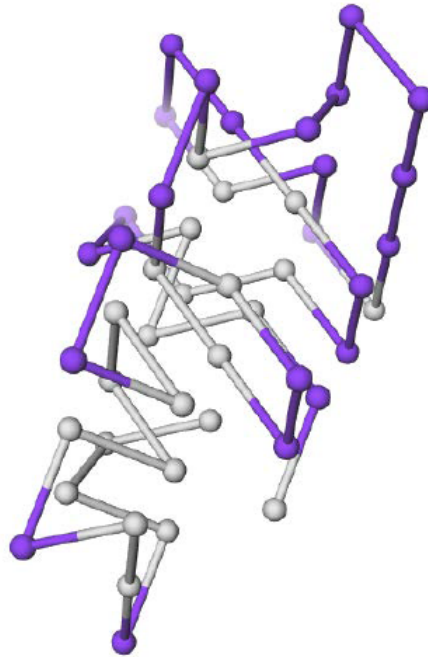


Figura 11: Conformação pelo modelo FCC

Fonte: Retirado de (Varela e Santos 2017) p.3.

A seguir o modelo 3DHPSC será explicado mais detalhadamente pois é objeto de estudo desta dissertação.

2.3.1.1 MODELO 3DHPSC

O modelo 3DHPSC (Li et al. 2002) apresenta uma metodologia onde as cadeias laterais (*Side Chains*) são introduzidas sobre o modelo 3DHP clássico. Ele busca melhorar as possibilidades de dobramento e uma maior reprodução das características de uma proteína.

No modelo de (Li et al. 2002) a energia livre considera a disposição espacial dos aminoácidos e da cadeia lateral. A equação 1 apresenta o calculo da energia livre proposta.

$$H = \epsilon_{bb} \sum_{i=1, j>1, i+1}^N \delta_{r_{ij}^{bb}, a} + \epsilon_{bs} \sum_{i=1, j \neq 1, i+1}^N \delta_{r_{ij}^{bs}, a} + \epsilon_{ss} \sum_{i=1, j>1, i+1}^N \delta_{r_{ij}^{ss}, a} \quad (1)$$

Onde os termos:

- ϵ_{bb} , ϵ_{bs} e ϵ_{ss} representam as ponderações de energia sobre cada tipo de interação possível,

ou seja, *backbone/backbone* (BB-BB) , *backbone/side-chain* (BB-SC), *side-chain/side-chain*(SC-SC);

- r_{ij}^{bb} , r_{ij}^{bs} e r_{ij}^{sc} são as distâncias no espaço tridimensional entre o i -ésimo e o j -ésimo resíduo das interações entre BB-BB, BB-SC e SC-SC respectivamente.
- O operador $\delta_{r_{ij},a}$ retorna 1 quando a distância entre o i -ésimo e o j -ésimo elementos é igual a constante a , caso o contrário retorna 0.

(Benítez 2010) propõe uma expansão do modelo proposto por (Li et al. 2002) visando melhorar o alcance de soluções da abordagem, levando em consideração a diferenciação dos tipos de cadeia lateral. O modelo de energia livre proposto é descrito na equação 2 a seguir.

$$\begin{aligned}
 H = & \epsilon_{HH} \sum_{i=1, j>1}^N \delta_{r_{ij}^{HH}, a} + \epsilon_{BB} \sum_{i=1, j>i+1}^N \delta_{r_{ij}^{BB}, a} + \\
 & \epsilon_{BH} \sum_{i=1, j \neq i}^N \delta_{r_{ij}^{BH}, a} + \epsilon_{BP} \sum_{i=1, j \neq i}^N \delta_{r_{ij}^{BP}, a} + \\
 & \epsilon_{HP} \sum_{i=1, j>i}^N \delta_{r_{ij}^{HP}, a} + \epsilon_{PP} \sum_{i=1, j>i}^N \delta_{r_{ij}^{PP}, a}
 \end{aligned} \tag{2}$$

Onde os termos:

- ϵ_{HH} , ϵ_{BB} , ϵ_{BH} , ϵ_{BP} , ϵ_{HP} e ϵ_{PP} representam as ponderações de energia sobre cada tipo de interação possível, sendo elas: cadeias laterais hidrofóbicas (HH), *backbone-backbone* (BB), *backbone-cadeia lateral hidrofóbica* (BH), *backbone-cadeia lateral Polar* (BP), cadeias laterais hidrofóbica-polar (HP) e cadeias laterais polares (PP).
- r_{ij}^{HH} , r_{ij}^{BB} , r_{ij}^{BH} , r_{ij}^{BP} , r_{ij}^{HP} e r_{ij}^{PP} são as distâncias no espaço tridimensional entre o i -ésimo e o j -ésimo resíduo das interações entre HH, BB, BH, BP, HP, PP respectivamente.
- O operador $\delta_{r_{ij},a}$ retorna 1 quando a distância entre o i -ésimo e o j -ésimo elementos é igual a constante a , caso o contrário retorna 0.

Com base na equação 2 descrita foi proposta uma função objetivo :

$$F = Energia * RadiusGh * RadiusGp \tag{3}$$

onde:

- F é a função objetivo

- Energia é um termo que leva em consideração a energia livre H penalizada pela quantidade colisões(NC) e um termo de peso para ponderação(PP). Sendo assim o termo corresponde :

$$Energia = H - (NC * PP) \quad (4)$$

- *RadiusGp* e *RadiusGh* representam o raio de giração das cadeias laterais polares e hidrofóbicas respectivamente. Esses termos serão detalhados mais adiante.

Os termos de raio de giração *RadiusGp* e *RadiusGh* dizem respeito a compacticidade da estrutura gerada. Sendo assim quanto mais compacta a estrutura for, menores os valores dos raios de giração.

A equação a seguir mostra como é calculado o raio de giração entre as cadeias laterais hidrofóbicas(*Rgh*):

$$Rgh = \sqrt{\frac{\sum_{i=1}^{n_h} (x_i + \bar{X}) + (y_i + \bar{Y}) + (z_i + \bar{Z})}{n_h}} \quad (5)$$

onde,

- x_i, y_i, z_i são as coordenadas de cada cadeia lateral do i-ésimo resíduo hidrofóbico da proteína;
- $\bar{X}, \bar{Y}, \bar{Z}$ são as médias de todos os x_i, y_i, z_i respectivamente;
- n_h é o número de resíduos hidrofóbicos da proteína.

O Termo *RadiusGh* é calculado sobre as coordenadas no espaço tridimensional dos resíduos hidrofóbicos, como apresentada na equação a seguir:

$$RadiusGh = MaxRGH - Rgh \quad (6)$$

O termo *MaxRGH* representa o pior caso a ser assumido, ou seja, o caso onde a proteína esteja totalmente esticada. O termo *RadiusGp* segue o mesmo princípio, porém leva em consideração o resíduos polares. A equação a seguir apresenta o calculo do raio de giração entre as cadeias laterais polares (*Rgp*).

$$R_{gp} = \sqrt{\frac{\sum_{i=1}^{n_p} (x_i + \bar{X}) + (y_i + \bar{Y}) + (z_i + \bar{Z})}{n_p}} \quad (7)$$

- x_i, y_i, z_i são as coordenadas de cada cadeia lateral do i -ésimo resíduo polar da proteína;
- $\bar{X}, \bar{Y}, \bar{Z}$ são as médias de todos os x_i, y_i, z_i respectivamente;
- n_h é o número de resíduos polares da proteína.

Porém a equação do $RadiusGp$ é diferente pois os resíduos polares tendem a ficar dispostos na parte externa da proteína em contato com o solvente biológico, gerando um núcleo hidrofóbico. Portanto o R_{gp} deve ser maior que o R_{gh} para que uma determinada conformação seja promissora. Logo o $RadiusGp$ é calculado seguindo a equação a seguir:

$$RadiusGp = \begin{cases} 1, & \text{se } (R_{gp} - R_{gh}) \geq 0 \\ \frac{1}{1 - (R_{gp} - R_{gh})}, & \text{caso contrário} \end{cases} \quad (8)$$

A seção a seguir apresenta brevemente os modelos *off-lattice* e seus algoritmos.

2.3.2 MODELOS OFF-LATTICE

Os modelos *off-lattice* são modelos com liberdade de rotação dos dobramentos sobre o plano tridimensional. Tendem a ser mais realistas em relação aos modelos em *lattice* e apresentam maior complexidade devido a inclusão de detalhes e a vasta gama de campos de força desenvolvidos. Esses modelos partem da premissa básica de que a estrutura nativa de uma proteína em um dado ambiente corresponde a energia livre mínima do sistema (hipótese termodinâmica) (Klepeis e Floudas 2003).

A tabela 1 apresenta uma lista de alguns algoritmos de modelagem *ab initio off-lattice*, são mostrados junto com suas funções de energia, métodos de busca conformacional, modelos e tempo de CPU (Lee et al. 2017).

Tabela 1: Lista dos principais Algoritmos

Algoritmo	Campo de Força	Método	Modelo	Tempo de CPU

AMBER/CHARMM/OPLS (Brooks et al. 1983; Weiner et al. 1984; Jorgensen and Tirado-Rives 1988; Duan and Kollman 1998; Zagrovic et al. 2002)	Physics- based	Dinâmica Molecular	Energia Mínima	Anos
UNRES (Liwo et al. 1999; Liwo et al. 2005, Oldziej et al. 2005)	Physics- based	CSA	Clustering/ Energia livre	Horas
ASTRO-FOLD (Klepeis et al. Klepeis and Floudas 2003; Klepeis et al. 2005)	Physics- based	BB/ CSA/ MD	Energia Mínima	Meses
ROSETTA (Simons et al. 1997, Das et al. 2007) http://www.rosetta.org	Physics and knowledge- based	Monte Carlo	Clustering/ Energia livre	Meses
TASSER/Chunk-TASSER (Zhang et al. 2004, Zhou and Skolnick 2007) http://cssb.biology.gatech.edu/ skolnick/webservice/MetaTASSER	Knowledge- based	Monte Carlo	Clustering/ Energia livre	Horas
I-TASSER (Roy et al. 2010; Yang et al. 2015a, b) http://zhanglab.ccmb.med.umich.edu/I- TASSER	Knowledge- based	Monte Carlo	Clustering/ Energia livre	Horas
QUARK (Xu and Zhang 2012) http://zhanglab.ccmb.med.umich.edu /QUARK	Knowledge- based	Monte Carlo	Clustering/ Energia livre	Horas

Fonte: Adaptado de (Lee et al. 2017)

AMBER/ CHARMM/OPLS : São métodos atualmente utilizados em diversas frentes na análise de biomoléculas. Consistem em conjuntos de campos de força moleculares, sendo esses campos de força disponíveis em domínios públicos e disponíveis para realização de simulações. Esses campos de força se integram com diversos tipos de pacotes para

simular dinâmica molecular, sendo estes pacotes específicos para diferentes biomoléculas a serem simuladas. Em relação as simulações são capazes de reproduzir sistemas biológicos e químicos em nível atômicos em escalas de tempo de femtossegundo a milissegundos (Salomon-Ferrer et al. 2013). O que diferencia os campos de força AMBER, CHARMM e OPLS em relação as propriedades moleculares simuladas, é no modo como os campos de força representados se relacionam as propriedades de átomos ligados (interações intramoleculares) e átomos não ligados (interações intermoleculares) (Jr et al. 2002, Jorgensen et al. 1996).

UNRES: O modelo UNRES (Liwo et al. 1997, Liwo et al. 1997, Liwo et al. 1998, Liwo 2013). é baseado em dinâmica molecular e apresenta uma abordagem tridimensional da disposição dos aminoácidos por uma modelagem *Coarse-Grained* (simplificações para modelagem molecular de biomoléculas) (Hadley e McCabe 2012). O UNRES é um modelo que realiza o dobramento no espaço espaço contínuo (*off-lattice*) e é *ab initio* ou seja, parte da premissa que é necessário apenas o conhecimento da sequência de aminoácidos para se obter a estrutura tridimensional da proteína. O modelo UNRES em sua implementação original utiliza o algoritmo *Conformational Space Annealing* (CSA) (Wilson e Cui 1994) (Lee et al. 1997) que é baseado no método de Monte Carlo mais popular, o algoritmo Manhattan (Baker et al. 1983). Logo a tendência é se explorar todas as possibilidades para se encontrar um ótimo global.

O modelo obteve bons resultados em algumas edições do *Critical Assessment of protein Structure Prediction* (CASP - Avaliação Crítica da Predição da Estrutura da Proteína), que é um evento que ocorre de dois em dois anos desde 1994 com o objetivo de avaliar os diversos métodos de predição de estrutura de proteínas, que fornece avaliações de métodos independente do estado da arte em modelagem de estruturas de proteínas (Dill et al. 2008). O CASP avalia os métodos de forma imparcial utilizando proteínas que são conhecidas pelos participantes até o início do evento, buscando manter não dar vantagens a nenhum grupo de pesquisa.

A figura 12 ilustra o modelo UNRES onde uma cadeia de polipeptídeos é representada por uma sequência de átomos de $C\alpha$ ligados por uma ligação virtual com as cadeias laterais (SCs) unidos ligados e grupos peptídicos unidos (p) localizados no meio entre os $C\alpha$ consecutivos. Os grupos peptídicos unidos são os centros de interações eletrostáticas de campo médio (ligações de hidrogênio), que determinam a arquitetura básica das cadeias polipeptídicas. Somente os grupos peptídicos unidos e SCs unidos servem como locais de interação. Os $C\alpha$ servem apenas para definir a geometria e não servem como locais de interação no modelo UNRES.

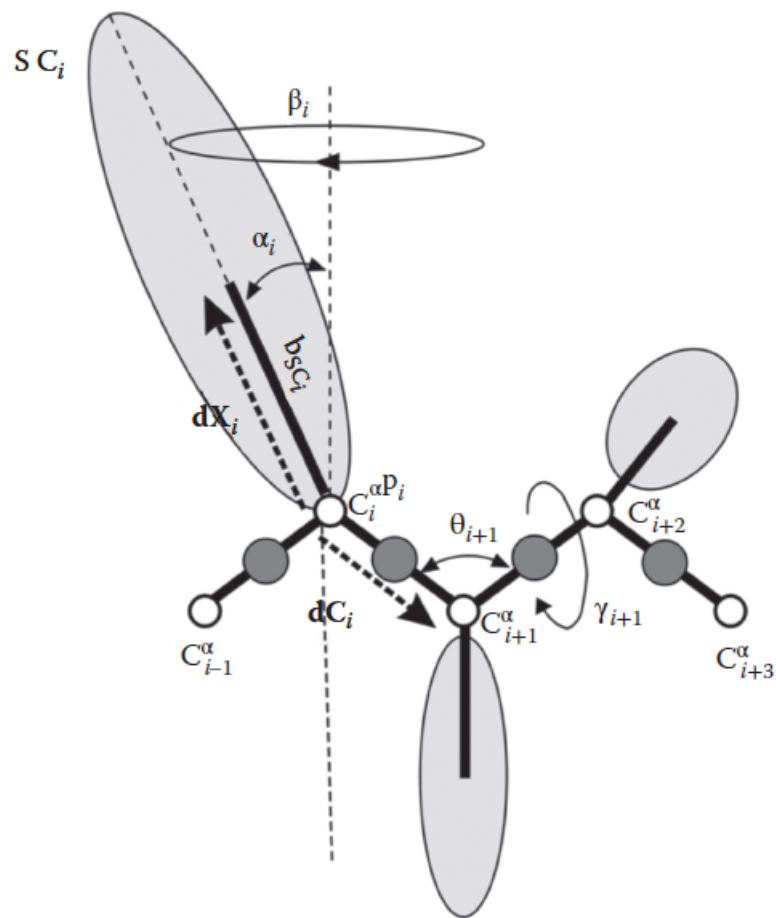


Figura 12: Modelo UNRES

Fonte: Retirado de (Papoian 2017) p.74.

ASTRO-FOLD: É um método que busca fidelidade a perspectiva fisicoquímica da predição de estruturas a partir da estrutura primária (Klepeis e Floudas 2003). O método é dividido em 4 estágios:

- O primeiro avalia as interações locais e a energia livre do sistema para encontrar locais de iniciação e término de α -hélices;
- O segundo avalia os contatos hidrofóbicos através de um modelo de hidrofobicidade para estimar superestruturas e prever a topologia de β -folhas na estrutura.
- O terceiro estágio deriva restrições no sistema com base nos estágios anteriores com os adicionais de cálculos para estimar os *loops* da estrutura;
- No quarto uma estrutura tridimensional é gerada utilizando otimização global determinística $\alpha\beta\beta$, otimização estocástica e dinâmica de ângulo de torção.

A figura 13 apresenta um panorama geral da abordagem e dos estágios descritos anteriormente.

ROSETTA: É um pacote de algoritmos para modelagem computacional e análises de estruturas de proteínas e macromoléculas em geral que considera fragmentos de estrutura secundária. Possui a vantagem de reduzir a entropia do espaço de busca conformacional e possui uma ampla variedade de algoritmos de amostragem eficazes para explorar o *backbone*, as cadeias laterais e o espaço das sequências. No método fragmentos curtos de proteínas conhecidas são montadas utilizando o método de Monte Carlo para produzir estruturas nativas utilizando apenas a informação da sequência primária (ROSETTA 2019, Rohl et al. 2004).

Tasser/Chunk-Tasser: O Tasser busca construir modelos completos e refinados de proteínas por meio de variações da simulação de Monte Carlo, empregando uma abordagem hierárquica que consiste na identificação de soluções por rearranjos, seguidos da montagem de estruturas terciárias, reorganizando-as em soluções no espaço contínuo (Zhang e Skolnick 2004). O Chunk-Tasser é uma variação do Tasser, consistindo em um modelo *ab initio* de estruturas tridimensionais de proteínas utilizando fragmentos de estruturas supersecundárias dobradas de um alvo para obter potenciais de contato e restrições de distância. Seu desempenho apresenta uma melhora significativa em relação a abordagem Tasser (Zhou e Skolnick 2007).

I-TASSER: É um servidor para predição de estruturas de proteínas e funções de proteínas. Propõe modelos 3D construídos com base em alinhamentos de múltiplos segmentos

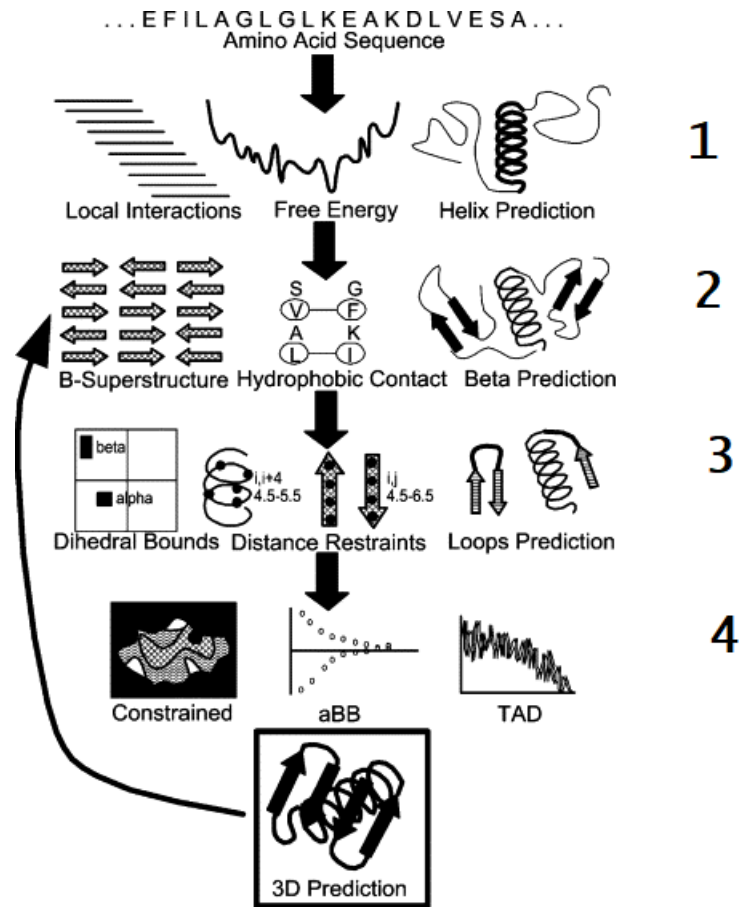


Figura 13: Astro-Fold

Fonte: Adaptado de (Klepeis e Floudas 2003)

de OMETS (Local Meta-Threading-Server)⁹ e simulações de montagem baseados em interações utilizando o Tasser clássico. As soluções obtidas são comparadas com bancos de dados de funções protéicas. Em constante desenvolvimento o I-TASSER conseguiu bons resultados nos experimentos teste dos CASP7, CASP8, CASP9, CASP10, CASP11, CASP12 e CASP13 (I-TASSER 2019, Yang et al. 2015).

QUARK: Prediz estruturas de proteínas *ab initio* e dobramento de peptídeos de proteínas, visando construir modelos de proteínas 3D corretamente. Constrói modelos a partir de pequenos fragmentos utilizando a simulação de Monte Carlo. É um modelo utilizado para prever estruturas de proteínas que não possuam modelos no PDB pois parte apenas da estrutura primária da proteína de entrada para prever sua estrutura. Foi classificado como o servidor número 1 em modelagem livre nos CASP9 e CASP10 (QUARK 2019).

⁹É um serviço online de geração de estruturas 3D de proteínas - disponível em <http://zhanglab.ccmb.med.umich.edu/LOMETS/>

Dois tipos de campos de força são utilizados nos modelos de predição *off-lattice*, sendo descritos a seguir (Zhang 2009):

Knowledge-Based este tipo de abordagem, como o nome sugere se baseia em estruturas já conhecidas e análises estatísticas sobre as mesmas. Para predizer proteínas geralmente se utiliza este modelo com abordagens de inteligência computacional com dados de proteínas resolvidas disponíveis no PDB. Utilizam-se características extraídas dessas proteínas de forma empírica na tentativa de predizer outras estruturas nativas e/ou secundárias.

Physics-Based são campos de força baseados em dinâmica molecular onde são considerados termos de comprimento de ligação, ângulos de ligação e torcionais, interações eletrostáticas e de Van Der Waals. Os termos levam em consideração os átomos e suas interações entre si e com o meio, baseados em mecânica quântica ou empacotamento de cristais.

De maneira geral, a abordagem *physics-based* tende a abstrair mais características do enovelamento e das tendências estruturais com a adição de heurísticas, como tendências de enovelamento, e a *knowledge-based* tende a conseguir maior semelhança nas estruturas nativas pois se baseia em outras estruturas semelhantes e dados estatísticos.

É importante salientar que ainda não foram encontrados campos de força que reproduzam com exatidão todas as características das proteínas e as tendências de enovelamento, como por exemplo as α -hlices e as β -folhas (Almeida et al. 2016). Recentemente um levantamento foi realizado sobre a tendência das abordagens *knowledge-based* obterem sucesso em predizer estruturas secundárias de proteínas, conseguindo predizer com grande acurácia as estruturas de α -hlices, β -folhas e β -turns, porém o levantamento apontou um gargalo nas demais estruturas, pois há indícios de que faltam informações no PDB para predição das demais estruturas (Yang et al. 2016).

Os métodos de busca utilizados nas abordagens *off-lattice* definem a acurácia e o tempo computacional na execução dos métodos, identificando mínimos locais e globais para cada tipo de função de energia empregado, gerando possibilidades de busca no espaço conformacional, e convergindo para estruturas estruturas com menor energia livre.

A Simulação de Monte Carlo (ou Método de Monte Carlo) (Hammersley 2013) é a abordagem mais difundida e utilizadas pelos em meios computacionais para análises de espaços de busca, vastamente utilizado nos mais variados tipos de problemas de otimização. O Recozimento Simulado (SA - Simulated Annealing) (Kirkpatrick et al. 1983) é a implementação da simulação de Monte Carlo Metropolis, no PSP consiste em gerar estados conformacionais de acordo com a distribuição de Boltzmann (Duhr e Braun 2006), e vai

convergindo para a solução conforme determinados parâmetros pré estabelecidos forem diminuídos, muitas vezes é utilizada a temperatura do espaço de busca como parâmetro.

O CSA (Lee et al. 1997) é um método que engloba a utilização do recozimento simulado e os algoritmos genéticos para melhorar o desempenho do algoritmo de busca. Consiste na aplicação de buscas de ótimos locais para a posterior aplicação do método de recozimento para encontrar as soluções. Tende a ser mais eficiente que o método clássico do *Simulated annealing* pois tende a eliminar buscas em espaço que possuem ótimos locais pouco relevantes (Joung et al. 2018).

A dinâmica molecular (MD - Molecular Dynamics) leva em conta as equações do sistema e tende a ser demasiadamente complexa para ser resolvida, mesmo para pequenas proteínas e de forma isolada sem a utilização de métodos de busca tende a ser muito demorada para convergir para o regime permanente. Porém é método mais utilizado para validar os campos de força pois demonstra as tendências das moléculas em convergir para o estado de energia livre mínimo e também auxilia na análise do comportamento das proteínas durante o dobramento (Duan e Kollman 1998).

O α BB (*Alfa branch And Bound*) é um método matemático, ou seja, não utiliza heurísticas ou modelos estocásticos para encontrar resultados. Para modelagem de proteínas é uma técnica eficiente no que diz respeito a encontrar soluções ótimas, porém possui elevado custo computacional, além de gerar várias soluções com muitos graus de liberdade (Klepeis e Floudas 2003).

Na próxima seção serão abordados alguns dos métodos de computação bioinspirados encontrados na literatura.

2.4 COMPUTAÇÃO BIOINSPIRADA

A computação bioinspirada é uma linha de pesquisa computacional que se desenvolveu com a observação dos comportamentos da natureza. Sendo que dados diversos padrões encontrados, foram reconhecidos seus padrões e anotados de forma a gerar algoritmos para resolver os diversos tipos de problemas existentes.

Atualmente a computação bioinspirada divide-se em duas grandes áreas, que são a Computação Evolucionária (CE) que baseia-se em estratégias com base na teoria da evolução, e a inteligência de Enxames (IE), se baseia nos comportamentos sociais de grupos de animais e insetos (Hattori et al. 2016). Dentre os principais métodos bioinspirados, podemos citar:

Algoritmos Genéticos (Holland 1975) : Método que parte do princípio da evolução natural buscando fornecer soluções de otimização baseando-se em um esquema de população de indivíduos que representam pontos de busca no espaço amostral do sistema analisado. É muito utilizado em sistemas onde se pode discretizar as variáveis, sendo possível também com variáveis contínuas.

Evolução Diferencial (Storn e Price 1997) é um método que evolui em um processo que envolve o cálculo de uma diferença ponderada entre dois vetores da população e posteriormente a sua adição a um terceiro vetor, considerando parâmetros como fator de mutação, cruzamento e estratégias de mutação. É um método semelhante ao algoritmo genético, porém tem sido mais eficaz para problemas em espaço contínuo.

Programação Genética (Koza 1990) Diferentemente da evolução diferencial e dos algoritmos genéticos a programação genética não limita o espaço de busca para encontrar as soluções. Os indivíduos da população são tratados como programas que varrem o espaço de busca e montam árvores sintáticas com soluções candidatas do problema. Essas soluções quando limites de execução ou solução são atingidos, recombina os ramos das árvores sintáticas de forma a apresentar a solução da busca.

Otimização do Lobo Cinzento (GWO - Grey Wolf Optimizer) é um algoritmo relativamente novo (2014) que imita a hierarquia e a forma de caça dos lobos cinzentos na natureza. Sobre hierarquia de liderança, organiza os lobos (soluções) entre alfa, beta, ômega e delta de acordo com a adequação das soluções. É um algoritmo eficiente e que possui poucos operadores envolvidos em seu funcionamento, sendo eles o número de agentes e o número de interações (Mirjalili et al. 2014).

Colônia Artificial de Abelhas (ABC - Artificial Bee Colony) : É um método inspirado no comportamento das abelhas em busca de alimento. Se baseia na auto-organização e divisão de trabalho entre as abelhas, como condições para atingir um comportamento coletivo inteligente. As abelhas procuram fontes de alimento e exploram com base na sua qualidade e distância. Utiliza uma base de exploração entre abelhas empregadas que realizam a busca local por alimento e uma exploração global realizada por abelhas escoteiras. É um método que possui a vantagem de utilizar poucos parâmetros, porém possui a limitação de suas soluções serem restritas ao espaço de busca das soluções iniciais (Karaboga e Basturk 2008).

Otimização de colônia de formigas (ACO - Ant Colony Optimizatio) : Como o próprio nome diz, é uma meta-heurística baseada na observação do comportamento das

colônias de formigas buscando alimento. As formigas criam caminhos de feromônio até encontrar o alimento (solução), sendo assim o algoritmo propõe a utilização de formigas (como agente computacional) caminhando pelo espaço de busca, adicionando feromônios conforme encontram soluções candidatas para o problema. A quantidade de feromônio para este método implica na chance de o caminho ser seguido para encontrar a solução (Dorigo et al. 1996, Dorigo e Birattari 2011).

Otimização de enxame de partículas (PSO - Particle Swarming Optimization) : É uma meta-heurística baseada no comportamento social de espécies que vivem em grupos na natureza como cardumes de peixes e o comportamento migratório de pássaros. O algoritmo é guiado pelas experiências de cada agente computacional para em busca das soluções. Os agentes são guiados no espaço de busca por dois termos, sendo eles a experiência local do agente (melhor solução do agente) e o outro a melhor experiência entre os outros agentes do enxame. Tende a ser um algoritmo eficiente na exploração do espaço de busca (extremos da superfície) pois utiliza muitos agentes (enxames) (Kennedy 1995).

Algoritmo do Morcego (BA - Bat Algorithm) (Yang 2010): É uma meta-heurística baseada no comportamento dos morcegos da espécie *Microchiroptera* na busca por alimentos e na sua habilidade de ecolocalização¹⁰. Os morcegos (agentes computacionais) voam aleatoriamente pelo espaço de busca em uma velocidade estipulada até uma posição qualquer, enquanto voam eles emitem sons com frequência, volume e uma taxa de emissão (pulso) fixos para explorar localmente o espaço. Conforme se aproximam de soluções a frequência dos pulsos e a taxa de emissão são reajustadas (a taxa de pulso é acelerada e a amplitude aumenta para melhorar a acurácia sobre a presa). A população de morcegos se move no espaço de busca continuamente atualizando a frequência, velocidade e posição de cada morcego. As buscas são realizadas pelos agentes até que os critérios estipulados sejam satisfeitos.

A seguir serão discutido os algoritmos genéticos mais detalhadamente pois, eles compõe grande parte dos algoritmos evolutivos multiobjetivos que serão descritos na seção 2.5.

¹⁰Capacidade dos morcegos de se orientar pela emissão de sons e percepção dos ecos resultantes. A percepção dos ecos faz com que os morcegos consigam estipular com grande precisão a localização e até o formato dos obstáculos do ambiente (Hutson e Mickleburgh 2001)

2.4.1 ALGORITMOS GENÉTICOS

Os algoritmos genéticos baseiam-se na teoria da evolução das espécies de Darwin, que diz respeito a espécies na natureza e seu comportamento com relação a adaptação ao meio ou extinção de espécies fracas ou inapropriadas. Os indivíduos fortes e adaptados a situação onde se encontram tem maior possibilidade de passar seus genes para as futuras gerações pela reprodução. No longo prazo as espécies que se adaptaram melhor ao meio e tiveram a melhor combinação de genes tornaram-se predominantes. Durante o processo de evolução alguns genes podem ser alterados aleatoriamente e se essa alteração for vantajosa esses indivíduos tendem a sobreviver, caso contrário serão eliminados pela seleção natural, assim como as novas gerações que evoluem a partir do material genético dos pais velhos, porém se essa transição não for suficiente a espécie estará ameaçada em algum momento pela seleção natural (Holland 1975).

Em um GA um vetor solução $x \in X$ é denominado de indivíduo ou cromossomo. Cromossomos são compostos por várias unidades chamadas de genes. Cada gene controla um ou mais recursos do cromossomo. Originalmente os genes propostos por (Holland 1975) são tratados como binários, posteriormente os genes foram tratados de diversas formas. Cada cromossomo é uma solução do espaço de soluções. O mecanismo de mapeamento entre o espaço de soluções e cromossomos é denominado codificação. O GA é implementado de forma a abordar a codificação de um problema, não o problema em si (Konak et al. 2006).

A população do cromossomo é aleatoriamente iniciada e a medida que a busca avança a população vai sendo ajustada a soluções que convirjam a soluções ótimas. Os operadores básicos de um GA são cruzamento (*crossover*) e mutação, sendo o *crossover* o operador mais importante. O *crossover* é responsável criação de novas populações conforme as gerações avançam. Ele seleciona indivíduos da população para cruzarem suas informações genéticas para gerarem filhos. A seleção dos indivíduos para *crossover* pode ser realizada de diversas formas, que serão exploradas mais adiante nesta seção. Este cruzamento visa garantir que os indivíduos da próxima geração sejam melhores geneticamente que os pais, ou seja, estejam mais próximos da solução ótima do que os indivíduos da população atual.

A mutação consiste em alterar aleatoriamente, no caso da codificação ser a binária, um ou mais bits de algum indivíduo da população, obedecendo uma probabilidade muito baixa para ocorrer. A mutação tem um papel importante no GA pois, em alguns casos o espaço de soluções tende a estagnar em algum ponto, e ela reintroduz a diversidade na população. Para avaliar o quão boa é uma população uma função *fitness*, também chamada por função objetivo deve ser presente no método. Ela é responsável por realizar a avaliação da aptidão dos indivíduos da população e dos filhos gerados. A cada geração o GA seleciona os indivíduos mais aptos da

população e dos filhos gerados para estarem presentes na nova geração, sendo os os com menor *fitness* descartados (Konak et al. 2006).

Os operadores do algoritmo do GA podem ser descritos da seguinte forma (Goldberg e Holland 1988):

População inicial : A geração da população inicial geralmente ocorre de forma aleatória. Quando se tem informações disponíveis sobre o problema ou deseja-se uma população com alguma características podem ser incrementados métodos para essa geração, como por exemplo iniciar a população com uma faixa de valores com *fitness* considerado bons. Deve-se tomar um cuidado especial com a geração desses indivíduos para problemas com restrições, para não serem gerados indivíduos inválidos nessa inicialização.

Seleção: Durante essa etapa são selecionados os pais de acordo com a aptidão avaliada pela função objetivo ou aleatoriamente. Para a seleção alguns métodos foram desenvolvidos, dentre os quais destacam-se:

Roleta: Neste método são avaliados todos os indivíduos e é aplicada uma probabilidade de serem sorteados de acordo com o seu aptidão. Então é gerado um numero aleatório onde é selecionado um indivíduo que estiver dentro desta faixa de probabilidade.

Uniforme: Todos os indivíduos da população tem a mesma probabilidade de serem selecionados.

Torneio: Os indivíduos são sorteados aleatoriamente, porém competem entre si para serem selecionados para cruzar. Esse método geralmente utiliza a aptidão para calcular quem é o pai mais apto para o cruzamento.

Para exemplificar, a figura 14 apresenta um exemplo com 4 indivíduos com suas probabilidade de acordo com cada método. É importante observar, que apesar de todos terem a mesma probabilidade no caso do torneio, os indivíduos sorteados competiriam entre si, logo, o indivíduo 2 seria o mais apto de todos no caso de ser selecionado, seguido pelo indivíduo 1, indivíduo 3 e 4 respectivamente.

Indivíduo	Aptidão	Roleta	Uniforme	Torneio
1	100	25%	25%	25%
2	200	50%	25%	25%
3	75	18,75%	25%	25%
4	25	6,25%	25%	25%
Total:	400	100%	100%	100%

Figura 14: Exemplo da distribuição nos tipos de seleção do algoritmo genético

Fonte: Autoria Própria

Crossover: É onde recombina-se as características dos pais selecionados para gerar uma nova população de filhos. O cruzamento pode ocorrer de diversas formas, para a codificação binária pode-se citar:

Ponto Único: Neste método é selecionado um ponto de corte em cada um dos pais, e o primeiro filho é gerado com a combinação entre a parte anterior a este ponto do primeiro pai com a parte posterior deste ponto do segundo pai. O segundo filho acaba por receber o inverso deste processo.

Múltiplos pontos: É feito de forma semelhante ao de ponto único, porém são escolhidas estratégias para intercalar os pontos selecionados na geração dos filhos.

Uniforme: Os genes a serem passados para os filhos são selecionados um a um de forma aleatória. Neste tipo de cruzamento os filhos tendem a ter maior aleatoriedade no seu material genético.

Mutação: A mutação modifica algumas características dos indivíduos selecionados com o objetivo de produzir variação genética. Para ocorrer a mutação uma probabilidade geralmente baixa é utilizada, onde um filho pode sofrer mutação. Das estratégias de mutação existentes para codificação binária, pode-se citar:

Swap: São trocados de posição um par ou mais de bits no material genético do indivíduo.

creep: São selecionados bits (um ou mais) do indivíduo e são somados ou subtraídos valores, de acordo com o critério escolhido.

flip : são determinados elementos para a substituir características do filhos sorteados.

Avaliação: É a etapa de cálculo de *fitness* onde são avaliados os filhos gerados para verificar quais são os mais aptos a estarem presentes na próxima geração. Nesta etapa pode-se

ordenar os pais e filhos para verificar quem são os mais aptos de acordo com a aptidão, para ser gerada a nova população.

Critério de Parada: Geralmente os GAs rodam por n gerações até retornarem a melhor solução encontrada. Porém em muitos casos quando os valores objetivos já são conhecidos ou estimados, pode-se adicionar critérios para a finalização do algoritmo, como por exemplo uma faixa de convergência. Este caso ocorre geralmente quando se está tentando avaliar a eficácia do algoritmo em problemas já conhecidos ou instancias *benchmarks*..

Formalmente um GA pode ser descrito pelos seguintes passos (Jones et al. 2002):

- 1 - Gerar as N soluções para a primeira população. Calcular a aptidão da população.
- 2 - Realizar o cruzamento e gerar os filhos. O cruzamento deve ser realizado com dois indivíduos x e y da população por algum critério de seleção.
- 3 - Realizar a mutação nos filhos de acordo com um parâmetro de mutação
- 4 - Calcular a aptidão dos filhos baseado na função objetivo.
- 5 - Selecionar N soluções com base nos pais e filhos gerados nos passos anteriores e adicionar a nova população até que seu tamanho seja satisfeito. As soluções que não estiverem entre as N soluções selecionadas são descartadas.
- 6 - Verificar se o número de gerações ou o critério de parada foi satisfeito. Se não foi, incrementa a geração e retorna a segunda etapa. Caso seja satisfeito retorna o melhor indivíduo.

O Pseudo código do GA básico é apresentado no **algoritmo 1** a seguir.

Algoritmo 1: Pseudo Código GA

Entrada: P, gen_{max}
Saída: Melhor_Individuo
1 início
2 Gerar_População_Aleatoria(P);
3 Avalia(P);
4 $gen_{atual} = 0$;
5 **repita**
6 $P' = \text{Seleciona_Pais}(P)$;
7 Crossover(P');
8 Mutação(P');
9 Avaliação(P');
10 $P = \text{Nova_Geração}(P, P')$;
11 Incrementa(gen_{atual})
12 **até** o critério de parada ou $gen_{atual} = gen_{max}$;
13 **retorna** Melhor_Individuo
14 fim

Na seção a seguir alguns dos algoritmos evolutivos multiobjetivo serão descritos, sendo que alguns são baseados em GAs.

2.5 ALGORITMOS EVOLUTIVOS MULTIOBJETIVO

Os algoritmos evolutivos tem sido utilizados na busca por soluções nos mais variados tipos de problemas de otimização, e também tem sido explorados para solucionar problemas multiobjetivos. Sendo assim, foram desenvolvidas técnicas para problemas multiobjetivos, denominadas na literatura como algoritmos evolutivos multiobjetivo (MOEA - *Multi-Objective Evolutionary Algorithm*) (TICONA 2003).

Problemas de otimização Multiobjetivos (MOOP - *Multi-Objective Optimization Problem*) possuem conjuntos de funções objetivo para serem otimizadas, com as funções contendo conjuntos de restrições a serem satisfeitas para se obter a solução ótima. Esses objetivos devem ser alcançados simultaneamente, o que inevitavelmente gera soluções conflitantes entre si. Tais conflitos dificultam a busca por soluções ótimas, pois ao melhorar uma função, uma ou mais funções podem ser influenciadas negativamente, ou seja, piorar (Deb 2011).

O problema multiobjetivo pode ser definido seguindo as seguintes restrições (Coello 2001):

maximizar/minimizar

$$f_m(x) \quad m = 1, 2, \dots, N_{obj}; \quad (9)$$

$$\text{restrito a } \begin{cases} g_i(x) \geq 0 & \text{com } i = 1, \dots, N_{R_{dsg}} \\ h_j(x) = 0 & \text{com } j = 1, \dots, N_{R_{igu}} \\ x_w^{(inf)} \leq x_w \leq x_w^{(sup)} & \text{com } w = 1, \dots, N_{var} \end{cases}$$

tal que $f_m(x)$ esta sujeita a $g_i(x)$, com $i = 1, \dots, N_{R_{dsg}}$, que são as restrições de desigualdade e $h_j(x)$, com $j = 1, \dots, N_{R_{igu}}$, que são as restrições de igualdade. As restrições $g_i(x) \geq 0$ e $h_j(x) = 0$ devem ser satisfeitas durante o processo de otimização (minimização/maximização) dos componentes do vetor $f_m(x)$, tal que x é um vetor das variáveis de decisão $x = (x_1, \dots, N_{obj})^t$. $x_w^{(inf)}$ e $x_w^{(sup)}$ são os limites inferiores e superior respectivamente, tendo o espaço de decisão limitado por esses valores. Uma solução é factível se x satisfaz as funções de restrições e o espaço de variáveis. Se não satisfizer essa condição, x não é factível. O conjunto de todas as soluções factíveis geram o espaço de busca (Brasil 2012).

O vetor de funções objetivo $f_m(x)$, em que N_{obj} é o número de objetivos, representa um espaço multidimensional chamado espaço de objetivos. Esse espaço é a principal diferença entre a otimização multiobjetivo e mono-objetivo. No espaço mono-objetivo o espaço de busca é unidimensional, sendo explorado apenas por uma função objetivo.

Vilfredo Pareto (1896) apresentou o conceito de dominância, posteriormente denominada Dominância de Pareto. Esse conceito apresenta noções de otimalidade e compara duas ou mais soluções factíveis do mesmo problema. Dadas duas soluções x e y , pode-se dizer que x domina y se as seguintes condições forem satisfeitas (Brasil et al. 2013):

- x é pelo menos igual a y em todas as funções objetivo;
- x é superior a y em pelo menos uma função objetivo.

Essas restrições definem um conjunto de soluções denominadas Pareto-ótimo, compostas por soluções não dominadas do conjunto de soluções. A figura 14 a seguir exemplifica graficamente combinações para duas funções objetivo, apresentando casos minimização de ambas as funções, maximização de f_1 e minimização de f_2 , maximização de f_1 e minimização de f_2 e maximização de ambas.

Formalmente a dominância de Pareto pode ser definida como (TICONA 2003):

Definição 1. *Uma solução x domina uma outra solução y se as condições seguintes forem satisfeitas:*

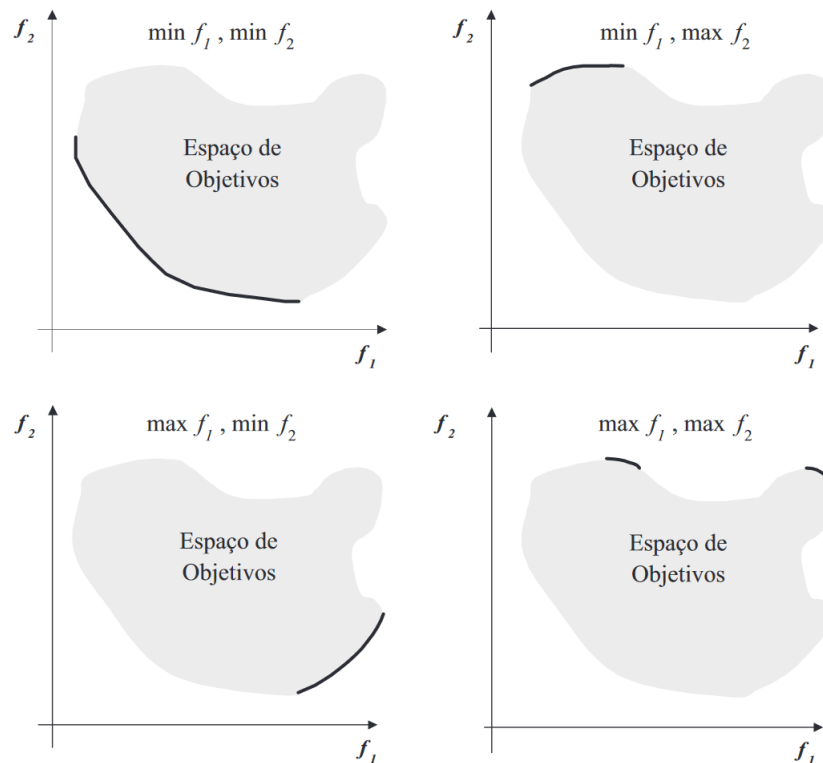


Figura 15: Exemplos de conjuntos de Pareto-Ótimo

Fonte: Retirado de (TICONA 2003) p.32.

1 - A solução x não é pior que y em todos os objetivos, ou seja, $f_m(x) \leq f_m(y)$ para todo $m = 1, 2, \dots, N_{obj}$.

2 - A solução x é estritamente melhor que y em pelo menos um objetivo, ou seja, $f_m(x) < f_m(y)$ pelo menos para um valor de m .

Se as condições da definição acima são satisfeitas, então:

1 - x é dominada por y ;

2 - y é não dominada por x ;

3 - x é não inferior a y .

Sendo assim, é possível comparar soluções com vários objetivos pelo conceito de dominância. A relação de dominância satisfaz as seguintes propriedades:

1 - Não é reflexiva, ou seja, uma solução não pode ser dominada por si mesma;

2 - Não é simétrica, ou seja, quando x domina y não implica que y domina x ;

3 - Transitiva, dado que se x domina y e y domina z , então x domina z .

Essas propriedades caracterizam a relação de dominância como uma relação de ordem parcial estrita (TICONA 2003).

O objetivo de um MOOP é identificar soluções no conjunto ótimo de Pareto, porém identificar todo o conjunto ótimo em muitos problemas multiobjetivos é praticamente impossível devido a sua complexidade e tamanho. Logo, uma abordagem de otimização multiobjetivo deve atingir os seguintes objetivos

Definição 2. *Um dado conjunto de soluções P , o conjunto não dominado P' é formado por:*

$$P' = \{x \in P \mid \nexists y : y \preceq x\} \quad (10)$$

Ou seja, quando um conjunto de soluções P corresponde ao conjunto de soluções factíveis de um MOOP, o o conjunto não-dominado P' é chamado de conjunto de Pareto ótimo. Seguindo esta definição surge a definição 3 onde é possível a existência de otimalidade local.

Definição 3. *Sendo P um conjunto de soluções e ε um número positivo arbitrariamente pequeno, o conjunto Pareto-ótimo local P'' é constituído por:*

$$P'' = \{x \in P \mid \nexists y : y \preceq x \wedge \|y - x\|_\infty \leq \varepsilon\} \quad (11)$$

Sendo assim a fronteira de Pareto para um MOOP pode ser definida como:

Definição 4. *Dado um MOOP com $F_m(x)$ e $m = 1 \dots N_{obj}$ funções por objetivo cujo conjunto Pareto ótimo é P' . A fronteira de Pareto PF é formada por :*

$$PF = \{f(x) \mid x \in P'\} \quad (12)$$

onde $f(x)$ é o vetor de funções objetivos para a solução x .

O que distingue os algoritmos evolutivos mono-objetivo dos MOEAs é o operador de seleção, onde a comparação entre as soluções deve-se realizar de acordo com o conceito de dominância de Pareto. Os MOEAs podem ser classificados em gerações (Brasil 2012) duas gerações, sendo:

Primeira Geração: Composta pelos Algoritmos VEGA (*Vector Evaluated Algorithm*) (Schaffer 1985), VOES (*Vector Optimized Evolution Strategy*) (Kursawe 1990), WBGA (*Weighted Based Genetic Algorithm*) (Hajela e Lin 1992), MOGA (*Multiple Objective Genetic Algorithm*) (Murata e Ishibuchi 1995), NSGA (*Non-dominated Sorting Genetic*

Algorithm) (Srinivas e Deb 1994), NPGA (*Niched-Pareto Genetic Algorithm*) (Horn et al. 1994).

No algoritmo VEGA foi proposta uma modificação do GA tradicional. É um método que trabalha com subpopulações e otimiza os objetivos separadamente. Ele propõe o cruzamento das populações resultantes de cada objetivo na busca por soluções próximas a região ótima de Pareto. O problema deste algoritmo é que ele tende as extremidades da fronteira de Pareto, por consequência, obtém pouca diversidade nas soluções encontradas. O VOES apresenta mecanismos para reter indivíduos não dominados e para excluir soluções excedentes. Esse critério de exclusão utiliza como critério a proximidade das soluções na fronteira de Pareto.

O WBGA aplica o conceito de pesos as funções objetivos e possui a vantagem de implementação simplificada e pode modelar funções de aptidão para problemas de minimização e maximização simultaneamente, utilizando o conceito de dominância de Pareto. O problema do WBGA é que em espaços de busca grande ele pode não ser eficiente em encontrar soluções ótimas, assim como outros métodos de busca baseados em pesos.

O MOGA é o primeiro algoritmo a ordenar soluções baseado em dominância, onde os indivíduos são ranqueados pela quantidade de indivíduos dominados por ele. Sendo assim os indivíduos não-dominados assumem valores altos com relação aos dominados, que são penalizados de acordo com a densidade da população onde estão inseridos.

Semelhante ao MOGA, o NSGA aplica o ranqueamento aos indivíduos da população. A diferença é que o NSGA classifica as soluções em classes e todas as soluções não dominadas de uma mesma classe recebem a mesma aptidão.

O NPGA aborda a questão de dominância de forma diferente, pois não calcula um valor de aptidão que prioriza soluções não dominadas. Ele insere um conceito de Torneio de Pareto onde a dominância é introduzida no operador de seleção. É um método que se torna vantajoso por não necessitar de um cálculo explícito para a função de aptidão e a complexidade não ser proporcional ao número de objetivos. Em contra partida novos parâmetros devem ser configurados e os mesmo influenciam no desempenho do algoritmo.

Segunda Geração: Desta geração destacam-se os algoritmos SPEA (*Strength Pareto Evolutionary Algorithm*) (Zitzler e Thiele 1998) e sua versão posterior SPEA2 (Zitzler et al. 2001), NSGA-II (Deb et al. 2000), PAES (*Pareto Archived Evolutionary Strategy*) (Knowles e Corne 1999), MOEA/D (*MultiObjective Evolutionary Algorithm Based on*

Decomposition) (Zhang e Li 2007).

O SPEA apresenta uma modelagem onde um arquivo de populações externas ao conjunto de soluções encontradas. Posteriormente seu sucessor SPEA2 propôs que diz respeito a distribuição das soluções sobre a fronteira de Pareto, ele utiliza um sistema de *Raw e Stregth* para calcular as melhores soluções em um arquivo, onde X e *Stregth* é quantidade de soluções dominadas por um indivíduo e o *Raw* é um somatório das *Stregth* das soluções que um indivíduo possui. O SPEA2 implementa uma estratégia de densidade de soluções sobre as soluções que vão para o arquivo também, objetivando garantir a eliminação de soluções próximas na fronteira de Pareto.

O NSGA-II apresenta melhorias em relação ao seu antecessor buscando o elitismo e implementando um conceito de *Crowding Distance*, onde a qualidade das soluções é avaliada de acordo com a distância entre as soluções não dominadas da fronteira de Pareto. Esta Abordagem será explorada mais adiante.

O PAES é descrito pelos autores como o algoritmo mais simples não trivial capaz de gerar boas soluções na cobertura do conjunto ótimo de Pareto. Consiste em um algoritmo que gera soluções aleatórias e para cada indivíduo realiza mutações e verifica se é um candidato apto, se for compara com os outros indivíduos e o insere em um conjunto de arquivo.

O MOEA/D é baseado na decomposição de objetivos em subproblemas escalares, onde são adicionados pesos a cada objetivo, realizando a busca pelos subproblemas gerados de forma paralela. A utilização de pesos diferentes torna possível encontrar diferentes pontes sobre a fronteira de Pareto. O MOEA/D é um algoritmo que tem se mostrado muito eficiente para problemas com muitos objetivos (Venske et al. 2014).

Na seção a seguir será melhor descrito o algoritmo NSGA-II que será utilizado neste estudo.

2.5.1 NSGA-II

O algoritmo é baseado na ordenação elitista por não dominância. Essa ordenação busca classificar as soluções em conjuntos de fronteiras $F_1 \dots F_n$ conforme o grau de dominância das soluções. A figura 16 ilustra essas fronteiras em um espaço de soluções hipotético (Deb et al. 2000).

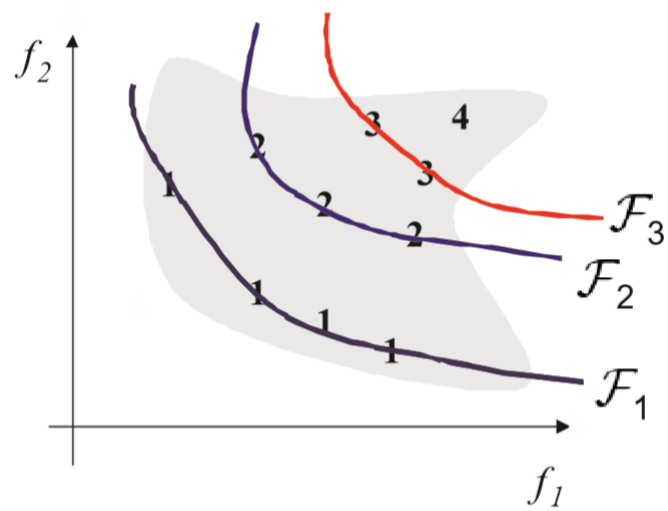


Figura 16: Exemplo de ordenação das Fronteiras por dominância

Fonte: Retirado de (TICONA 2003) p.75

A fronteira F_1 é composta por soluções não dominadas, a fronteira F_2 é composta por soluções dominadas apenas por F_1 e F_3 é composta por soluções dominadas por F_1 e F_2 .

O algoritmo 2 apresenta pseudo código do esquema de ordenação por dominância (Brasil 2012):

Algoritmo 2: ALGORITMO PARA ORDENAÇÃO POR DOMINÂNCIA

Entrada: Conjunto de soluções M

Saída: conjunto de fronteiras $F_1, F_2 \dots F_k$

```

1 início
2   para cada solução $_i \in M$  faça
3      $nd_i = 0$ ;
4      $U_i = \emptyset$ ;
5     para cada solução $_j \neq$  solução $_i$  e solução $_j \in M$  faça
6       se  $i$  domina  $j$  então
7          $U_p = U_p \cup j$ 
8       fim
9       se  $j$  domina  $i$  então
10         $nd_i = nd_i + 1$ 
11      fim
12    fim
13    se  $nd_i = 0$  então
14       $F_1 = F_1 \cup i$ 
15    fim
16  fim
17   $k = 1$ 
18  repita
19     $Temp = \emptyset$ 
20    para cada solução $_i \in F_k$  faça
21      para solução $_j \in U_i$  faça
22         $nd_j = nd_j - 1$ 
23        se  $nd_j = 0$  então
24           $Temp = Temp \cup j$ 
25        fim
26      fim
27    fim
28     $k = k + 1$ 
29     $F_k = Temp$ 
30  até  $F_k = \emptyset$ ;
31  retorna  $F_1, F_2 \dots F_k$ 
32 fim

```

Onde:

- M é o conjunto de soluções ;
- $F_1, F_2 \dots F_k$ são as fronteiras que foram classificadas pelo algoritmo;
- nd_i é a quantidade de soluções que dominam a solução i e
- U_i é o conjunto de solução que são dominadas pela solução i

As soluções com $nd_i = 0$ estão alocadas em F_1 até que o primeiro laço do algoritmo 2. Depois fim do primeiro laço o algoritmo, o segundo laço percorre o conjunto de soluções U_i para cada solução i de F_1 . O contador nd_j de cada solução j em U_i é decrementado em 1. Se $nd_j = 0$ então a solução j pertence a próxima fronteira. A processo é repetido até que todas as soluções estejam classificadas em fronteiras. O Algoritmo opera com duas populações denominadas P e Q de tamanho $N_{individuos}$ (Brasil 2012).

As populações serão denotadas por P_t e Q_t para simbolizar as passagens de gerações, onde $t = 1, 2, \dots, N_{iteracoes}$. Na primeira geração os indivíduos iniciais da população P_1 geram as soluções Q_1 pela aplicação dos operadores padrões de um GA. Na sequência é estabelecido um processo competitivo para preencher $N_{individuos}$ vagas na população posterior (P_{t+1}) entre $2N_{individuos}$ contidos em $R_t = P_t \cup Q_t$. Esta operação é realizada usando ordenação por dominância em R_t , encaminhando as soluções não-dominadas contidas nas fronteiras diretamente para a próxima geração, caracterizando o elitismo (Brasil 2012).

Em busca de garantir a diversidade das soluções perante a fronteira calculada, um método de estimativa a densidade das soluções que rodeiam cada indivíduo é utilizado. Este método chamado ordenação de distância de multidão (*Crowding Distance Sorting*) (Deb 2011), calcula a média das entre a distância entre duas soluções adjacentes a cada indivíduo, esse cálculo é realizado para todos os objetivos. O algoritmo 3 apresenta o procedimento para a realização do cálculo da distância de multidão para cada indivíduo (Brasil 2012) onde:

- M é o conjunto de soluções
- $crowdist_n$ é o valor da distância de multidão do n -ésimo indivíduo do conjunto M
- f_m é o valor da m -ésima função objetivo para o indivíduo
- $crownd_n$ são as distâncias de multidão da n -ésima solução em M

Algoritmo 3: CALCULO DA DISTÂNCIA DE MULTIDÃO

Entrada: Conjunto de soluções M
Saída: $Crowdist_i$; Valores de distância de multidão da n -ésima solução em M

```

1 início
2   para cada solução  $i \in M$  faça
3      $dist_i = 0$ 
4   fim
5   para  $m = 1, 2, \dots$  até  $N_{obj}$  faça
6     Classificar  $M$  por  $F_m$ 
7      $crowdist_1 = crowdist_M = \infty$ 
8     para  $i = 2$  até  $M - 1$  faça
9        $crowdist_i = crowdist_i + f_m(M_{i+1}) - F_m(M_{i-1})$ 
10    fim
11  fim
12 fim

```

A figura 17 ilustra as distâncias entre o indivíduo i e seu antecessor $i-1$ e o sucessor $i+1$ em uma fronteira hipotética. O seja cada individuo é classificado pelos n indivíduos que fazem vizinhança com ele. Este conceito é baseado no método KNN (*K-Nearest-Neighbours*) (Guo et al. 2003) que propõe um método simples e eficiente de calcular e classifica distâncias entre pontos em modelos de dispersão de dados. No KNN um dado ou individuo é classificado de acordo com a distância dos seus K -vizinhos, se ele se encontra próximo de algum determinado grupo ele é classificado neste grupo. Para o NSGA-II o método é eficiente principalmente para a formação da primeira fronteira, onde os indivíduos não são dominados.

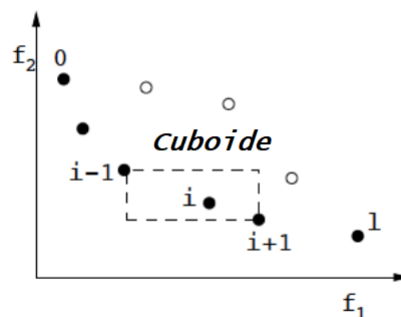


Figura 17: Distância de multidão de um ponto

Fonte: Adaptado de (Deb 2011) p.9

A determinação das distâncias de multidão ajudam na determinação das soluções que estão presentes na próxima geração, pois possibilita que soluções mais espalhadas na fronteira

de Pareto sejam selecionadas.

O NSGA II clássico faz utilização do processo de seleção por torneio. Porém como trata de mais de um objetivo ele apresenta diferenças do GA clássico. Logo, a seleção de uma solução i sobre uma j é realizada se:

- i possui um ranqueamento menor que j , ou seja, $rank_i < rank_j$
- Se ambas as soluções possuem o mesmo ranqueamento e i possui um maior valor de distância de multidão

A população Q_{t+1} é gerada utilizando a seleção por torneio, cruzamento e mutação na população P_{t+1} . O NSGA-II opera pelas $N_{iteracoes}$ e as soluções finais estão em $P_{N_{iteracoes}} \cup Q_{N_{iteracoes}}$. A figura 18 apresenta um panorama geral de como os conjuntos de soluções são selecionados para as próximas gerações de acordo com a fronteira a qual pertencem.

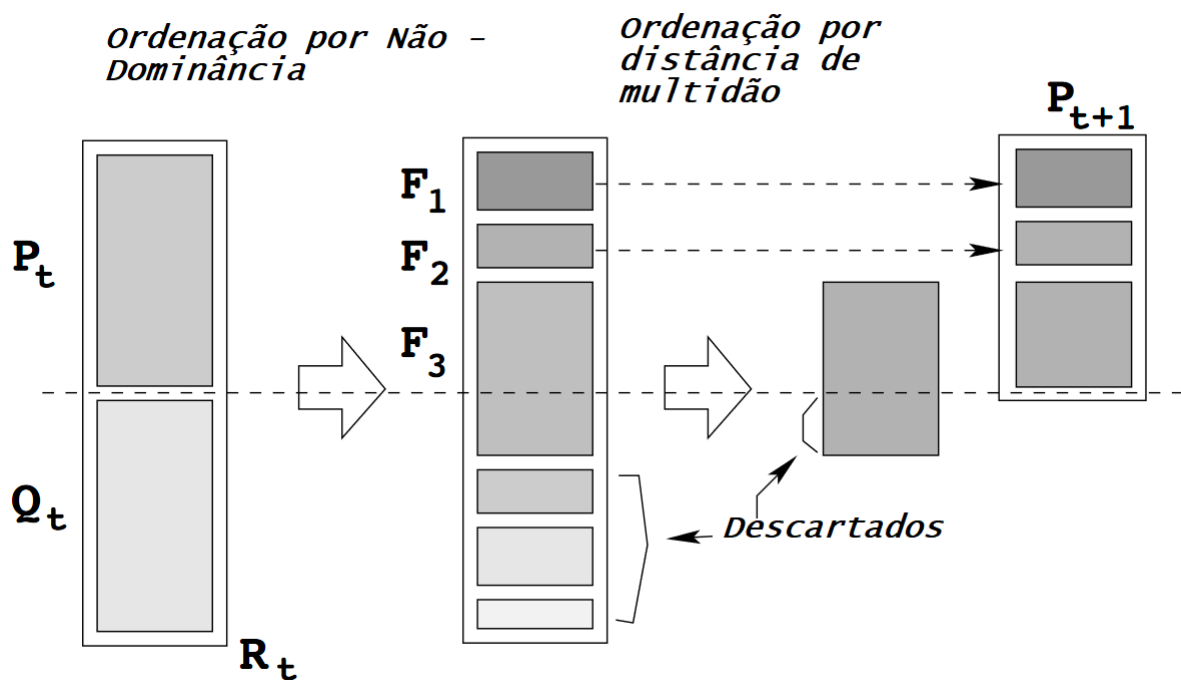


Figura 18: Panorama geral do NSGA-II

Fonte: Adaptado de (Deb 2011) p.9

Por fim, o algoritmo do funcionamento do NSGA II é apresentado a seguir.

Algoritmo 4: PSEUDO CÓDIGO DO ALGORITMO NSGA II

Entrada: Parâmetros do NSGA II

Saída: Soluções $P_{N_{iteracoes}} \cup Q_{N_{iteracoes}}$

```

1 início
2   Criar população inicial aleatória  $P_1$  de  $N_{individuos}$ 
3   Ordenar  $P_1$  por dominância
4   Aplicar operadores genéticos em  $P_1$  para gerar uma nova população  $Q_1$  de
   tamanho  $N_{individuos}$ 
5   para Geração  $t = 1, 2, \dots$  até  $N_{iteracoes}$  faça
6     Aplicar Ordenação por dominância em  $P_t$  e  $Q_t$ 
7      $k = 1$ 
8     repita
9       Realizar calculo da Distância de Multidão em  $F_k$ 
10       $P_{t+1} = P_{t+1} \cup F_k$ 
11      incrementa k em 1;
12     até  $P_{t+1} + F_k > N_{individuos}$ ;
13     Realizar calculo da Distância de Multidão em  $F_k$ 
14     Classificar  $F_k$  pelo ranqueamento e a distância de multidão
15     Receber as N melhores soluções de  $F_1, F_2, \dots, F_n$  em  $P_{t+1}$ 
16     Gerar nova população  $Q_{t+1}$  aplicando os operadores genéticos.
17   fim
18   retorna  $P_{N_{iteracoes}} \cup Q_{N_{iteracoes}}$ 
19 fim

```

2.6 TRABALHOS RELACIONADOS

Diversos métodos e modelos foram utilizados com o passar dos anos para realizar a predição de estruturas de proteínas. Nesta seção serão citados alguns trabalhos relacionados ao PDP.

(Cutello et al. 2005) utiliza o algoritmo PAES para o PDP sobre uma abordagem multiobjetivo onde são consideradas energias entre átomos ligados e não-ligados como critérios a serem minimizados.

(Li 2007) propõe a utilização do recozimento simulado (*Simulated Annealing*) em comparação com métodos da mesma natureza como o algoritmo de Monte Carlo(MC) e o algoritmo Evolutivo de Monte Carlo(EMC, Evolutionary Monte Carlo). O método proposto foi mais preciso nos modelos 3DHP E 2DHP.

(Thachuk et al. 2007)propoe o algoritmo Replica Exchange Monte Carlo(REMC) para o modelo HP e obtém bons resultados para estruturas complexas em 2D e 3D.

(Benítez e Lopes 2010) utiliza o algoritmo da colonia de abelhas para explorar o modelo 3DHPS com duas implementações diferentes: O método mestre-escravo e o método hierárquico híbrido.

(Mansour et al. 2012) faz o estudo do PSP sobre o modelo HP utilizando PSO, neste estudo é são realizados vários experimentos com parâmetros diferentes e comparados com a literatura, obtendo bons resultados com relação as estruturas geradas.

(Benítez e Lopes 2012) realiza estudo do modelo 3D AB *off-lattice* e apresenta algumas sequências *benchmarks* para estudo do modelo.

(Garza-Fabre et al. 2012) faz uma análise do modelo HP realizando a decomposição da função objetivo tradicional em duas funções objetivo. É realizada uma análise sobre os modelos 3D e 2D utilizando algoritmos evolucionários.

(Gabriel et al. 2012) propõe a utilização de um Algoritmo Evolutivo Multiobjetivo em Tabelas (AEMT) para o PDP, e obteve resultados inferiores aos da literatura para o modelo HP, porém conseguiu maior velocidade e robustez no que se refere a execução do algoritmo.

(Brasil et al. 2013) Propõe uma abordagem com uma expansão do AEMT. O Algoritmo Evolutivo Multiobjetivo em Tabelas (AEMMT)(em inglês, *Multiobjective*

evolutionary algorithms with many tables (MEAMT)) é implementado visando solucionar limitações com de problemas com vários objetivos. Para o PDP obteve bons resultados na predição de *beta*-folhas e estruturas relativamente complexas.

(Venske 2014) obteve bons resultados para o dobramento utilizando o algoritmo de evolução diferencial para o campo de força CHARMM, utilizando uma abordagem multiobjetivo. Foram comparadas as estruturas computacionais com estruturas reais e *benchmarks* e os resultados se mostraram promissores.

(Garza-Fabre et al. 2016) realizou um estudo utilizando um algoritmo memético que incorpora informações do dobramento ao GA e alguns métodos como buscas locais baseadas no ROSETTA.

(Traykov et al. 2016) apresenta a utilização de programação inteira para o estudo do modelo 2D hp e compara com outras meta-heurísticas da literatura, obtendo bons resultados.

(Wang et al. 2017) realiza um estudo sobre predição de dobramento de proteínas de membrana utilizando *deep learning*. Esta estudo teve bons resultados quando comparado com outros métodos no Continuous Automated Model EvaluatiOn (CAMEO).

(TRAYKOV et al. 2018) Propõe a utilização de programação inteira para estudo do modelo 3DHP. Neste estudo ele obtém resultados promissores se comparado com métodos outros métodos da literatura.

(Hattori et al. 2018) apresenta uma nova abordagem para PSP utilizando o método *deep learning Long Short-Term Memory* (LSTM) em um modelo *coarse-grained* 3D AB- *off-lattice*. Os resultados apresentaram que o algoritmo é promissor e capaz de aprender transições de dobramentos de proteínas.

(Todorin et al. 2018) propõe a utilização de programação não linear em um algoritmo de força bruta, e conseguiu bons resultados para o modelo 3D HP *off-lattice*.

(Chou et al. 2019) apresenta uma modificação no algoritmo *branch and bound* para o modelo 3DHP(denominado 3D-NBB), neste estudo o método obteve resultados computacionais satisfatórios sobre outros no que diz respeito a o tempo computacional.

(Senior et al. 2020) apresenta sistema AlphaFold que utiliza *deep-learning* para realizar predição de estruturas. O Alphafold obteve grande sucesso em prever estruturas mesmo com poucas informações de estruturas homólogas sendo utilizadas.

(Hattori et al. 2020) apresenta o estudo do modelo 3DHPSC utilizando programação inteira e sequencias reais obtidas no PDB e encontra soluções ótimas no estudo de pequenas

sequências, apresentando também a tendência exponencial do processamento e os desafios para sequências maiores .

3 METODOLOGIA

O modelo 3DHPSC estudado por (Benítez 2010) utiliza apenas uma função objetivo para explorar o espaço de busca de soluções, com o objetivo de estudar novas possibilidades e expandir a abordagem, este capítulo apresenta a decomposição da função *fitness* utilizada por (Benítez 2010) em duas, tornando o problema bi-objetivo e a descrição do algoritmo NSGA-II aplicado ao problema de dobramento de proteínas com o modelo 3DHPSC. Serão descritos os operadores do algoritmo e as funções utilizadas para a execução dos experimentos.

3.1 PROPOSTA BIOBJETIVO PARA O MODELO 3DHPSC

Este trabalho objetiva o estudo da expansão da função objetivo apresentada na equação 4 proposta por (Benítez 2010). A função foi decomposta em 2 funções objetivo, a primeira unicamente baseada na energia livre considerando a quantidade de contatos hidrofóbicos (f_1) e a segunda baseada na equação 4 (f_2).

Logo as funções objetivo deste trabalho são:

$$f_1 = HNC \quad (13)$$

$$f_2 = -(Energia * RadiusGh * RadiusGP) \quad (14)$$

Sendo que *HNC* (Número de contatos hidrofóbicos) deve ser maximizado buscando formar o núcleo hidrofóbico e f_2 deve ser minimizada, seguido a premissa da hipótese termodinâmica de Anfinsen (Anfinsen 1973), que supõe que a energia livre de uma proteína deve diminuir conforme ela converge para o seu estado nativo.

Inicialmente deve-se propor uma codificação para a população. Definiu-se que a codificação seria de coordenadas relativas ao movimento da proteína no espaço tridimensional,

ou seja coordenadas absolutas . Considera-se que cada elemento da proteína pode realizar uma quantidade finita de movimentos, pois o movimento da cadeia lateral depende da localização e orientação do *backbone*.

Para este trabalho considerou-se que inicialmente a primeira cadeia lateral estará nas coordenadas $x = 0, y = -1$ e $z = 0$ do plano cartesiano tridimensional e seu respectivo *backbone* estará na origem. Esta estratégia foi utilizada apenas para poder nortear o início da movimentação dos demais aminoácidos.

Os movimentos que podem ser realizados por uma cadeia lateral (SC - *de Side Chain*) serão relativos ao *backbone* que estiver conectado a ela, e o movimento de cada *backbone* será relativo ao *backbone* anterior.

Cada estrutura pode realiza apenas cinco movimentos em uma treliça quando referenciados ao seu antecessor (Frente, Cima, Baixo, Esquerda e Direita). Para padronizar esses movimentos este trabalho propõe que um vetor de referência deva ser criado contendo informações para o BB e a SC. Neste vetor de referência são armazenadas as orientações locais que a estrutura que está sendo montada tem disponível de acordo com a coordenada absoluta do *backbone* anterior.

O vetor de referência ($Vetor_{Ref}$) possui 8 bits, onde os cinco primeiros bits indicam com valores entre 1 e -1, se uma determinada coordenada no espaço tridimensional em relação ao *backbone* anterior cresce ou decresce. Os 3 últimos bits indicam o que seriam os eixos "frente", "lados" e "Cabeça-Pé". Essa estratégia foi utilizada para a estrutura ser norteadada conforme a montagem for ocorrendo. Para cada BB este vetor de referência é atualizado, mais adiante um exemplo da atualização do vetor e da movimentação será explorado.

Para codificar os possíveis movimentos que o BB (**F**rente, **C**ima, **B**aixo, **E**squerda, **D**ireita) e a SC (**f**rente, **c**ima, **b**aixo, **e**squerda, **d**ireita), optou-se pela combinação desses movimentos para a geração do cromossomo. Cada elemento do cromossomo é referente a um par BB-SC.

A tabela 2 a seguir representa o conjunto de movimentos relativos para codificação.

Com base nesses movimentos cada indivíduo do cromossomo proposto terá o tamanho total sequencia de aminoácidos -1, pois o primeiro par BB-SC sempre é alocado na mesma posição. Cada movimento é decodificado e representado por 5 bits. Sabendo-se que a representação de 5 bits consegue representar números de 0 a 31, uma estratégia de mudança de bits foi proposta. Esta estratégia consistem em selecionar aleatoriamente um ou os dois

Tabela 2: Movimentos relativos BB-SC

Movimentos	Esquerda BB	Frente BB	Direita BB	Baixo BB	Cima BB
esquerda-SC	0	5	10	15	20
direita SC	1	6	11	16	21
frente SC	2	7	12	17	22
baixo SC	3	8	13	18	23
cima SC	4	9	14	19	24

bits mais significativos e alterar seu valor para zero. A figura 19 apresenta um exemplo da codificação e decodificação utilizada na implementação do algoritmo.

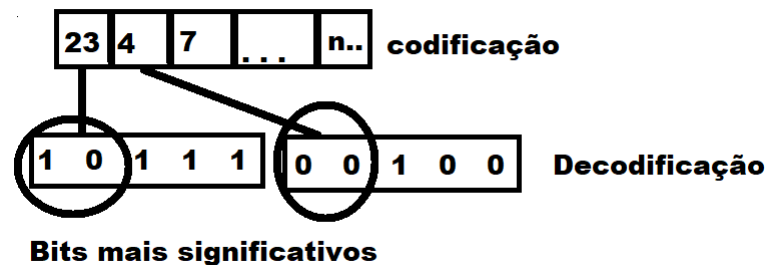


Figura 19: Exemplo da codificação e decodificação dos indivíduos

Fonte: Autoria Própria

O algoritmo 5 se descreve o processo referente ao mapeamento Genótipo→fenótipo da estrutura no espaço 3D:

Algoritmo 5: Algoritmo de Mapeamento Genótipo→fenótipo

Entrada: *Individuo*

Saída: *Matriz_{x,y,z}*

```

1 início
2   Posiciona o primeiro Par BB-SC na Matrizx,y,z
3   Monta Vetorref
4   para individuoi, comi = 1, 2... até individuomax faça
5     Posiciona o par BB-SC de acordo com movimento individuoi e Vetorref
        na Matrizx,y,z
6     Atualiza Vetorref
7   fim
8 fim
```

onde,

- *Individuo* é o vetor com os movimentos codificados

- $Matriz_{x,y,z}$ é uma matriz de tamanho sequencia de aminoácidos*2 (para poder comportar BB's e SC's) por 3 (para as coordenadas tridimensionais x,y e z).
- $individuo_i$ é o movimento local da interação e $individuo_{max}$ é o ultimo elemento do vetor de movimentos.
- $Vetor_{ref}$ é o vetor com as orientações dos movimentos possíveis e os sentidos dos eixos.

Para exemplificar os movimentos uma estrutura "HPH" será utilizada. Sabendo que o primeiro *backbone* é localizado na origem e a primeira cadeia lateral "H" será inserida no ponto $x = -1, y = 0$ e $z = 0$ tem-se a seguinte representação:

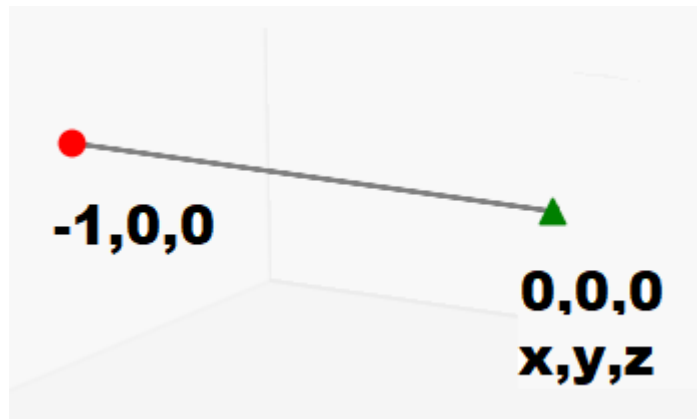


Figura 20: Primeiro backbone com sua respectiva cadeia lateral

Fonte: Autoria Própria.

Supondo os movimento 24 e 9 termos de acordo com a tabela 2 os movimentos BB-SC Cima-cima e Frente-cima. Padronizou-se que o $Vetor_{ref}$ é iniciado com a seguinte sequencia:

F	C	B	E	D	Lados	Frente	Cabeça-Pé
1	1	-1	-1	1	2	0	1

Onde

- Lados inicialmente tem o valor 2 que significa que é o eixo Z;
- Frente inicialmente tem o valor 0 que significa que é o eixo X e
- Pé inicialmente tem o valor 1 que significa que é o eixo Y.

Conforme a estrutura vai se movimentando os valores de $Vetor_{ref}$ vão se alterando, o 3 últimos termos do vetor auxiliam a orientação para os próximos movimentos do *backbone*. Sendo assim a orientação ficaria como a representada na figura 21

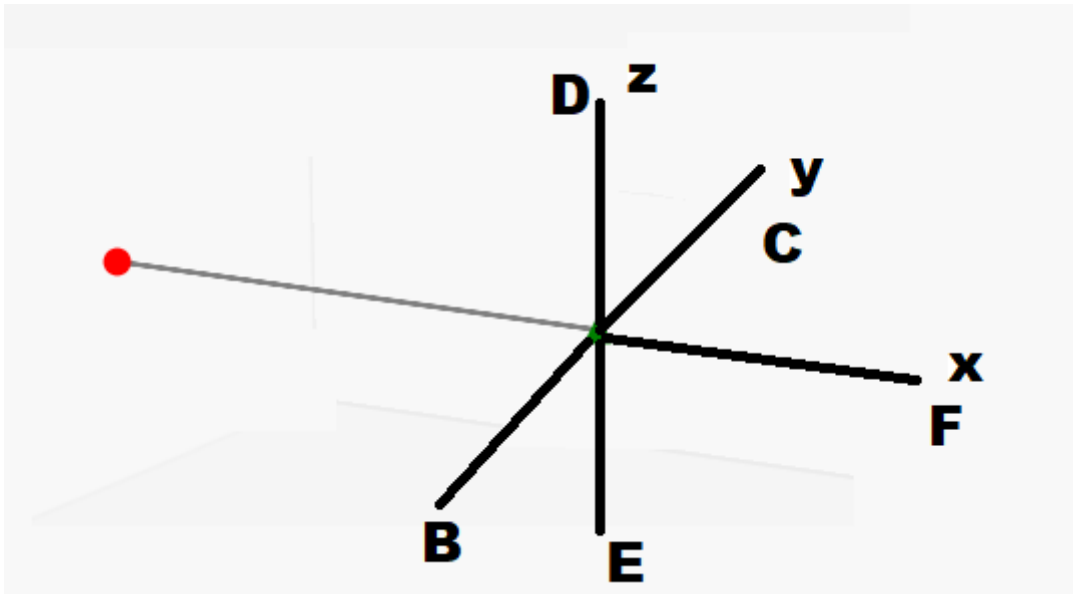


Figura 21: Orientação da primeira estrutura

Fonte: Autoria Própria.

Sendo assim, a segunda estrutura BB-CL a surgir estará em Cima-cima, ele vai se movimentar para cima no eixo Y, visto que ele é o eixo "Cabeça-Pé" no momento, como representado na figura 22.

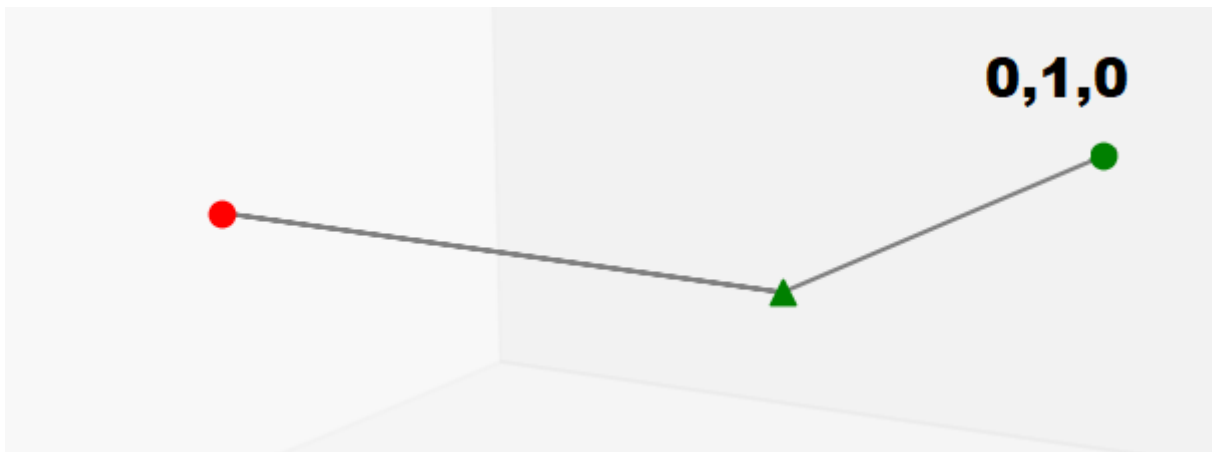


Figura 22: Orientação do segundo *backbone*

Fonte: Autoria Própria.

A posição 2 do $Vetor_{ref}$ inicial indica a soma de 1 ao valor da posição movimentada logo, saímos de 0,0,0 para 0,1,0 como apresentado na figura 22. Com este movimento a orientação é alterada e $Vetor_{ref}$ fica da seguinte forma:

Sendo assim, o eixo y passa a ser a "frente" de referência e o eixo X passa a ser o eixo

F	C	B	E	D	Lados	Frente	Cabeça-Pé
1	-1	1	-1	-1	2	1	0

”Cabeça-Pé”. Com essa nova orientação, a cadeia lateral deverá ir para cima, onde conta na tabela que deve ser decrescido -1 em X.

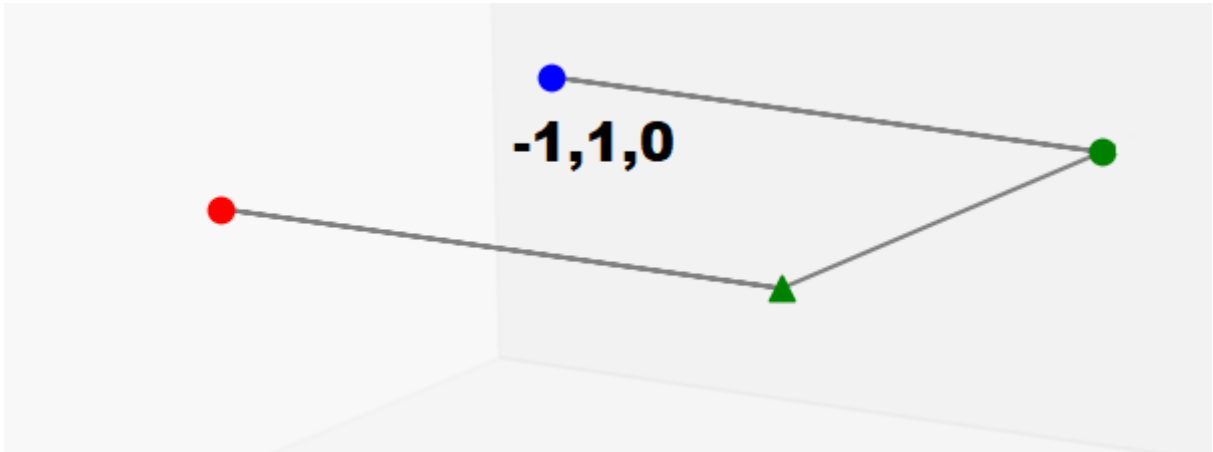


Figura 23: Orientação da segunda Cadeia Lateral relativa ao segundo *backbone*

Fonte: Autoria Própria.

Com base no segundo *backbone* o terceiro conjunto de movimentos é realizado (Frente-cima.). Sendo assim, é possível notar que o $Vetor_{ref}$ não é alterado quando o movimento do *backbone* é para frente, apenas quando é indicada uma rotação. Sendo assim, o próximo *backbone* é posicionado na posição 0,2,0 como mostrado na figura 24.

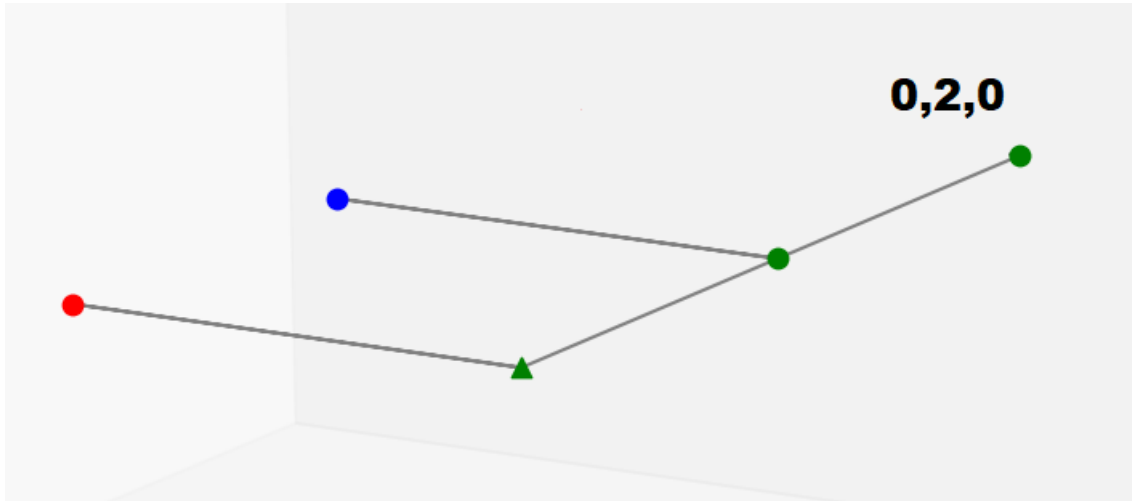


Figura 24: Orientação do Terceiro *backbone*

Fonte: Autoria Própria.

Por fim, o próximo movimento da cadeia lateral se dará na posição $-1,2,0$ pois ela se encontra para cima, $Vetor_{ref}$ indica que -1 é somado ao valor da posição atual. Ficando ficando representado como na figura 25.

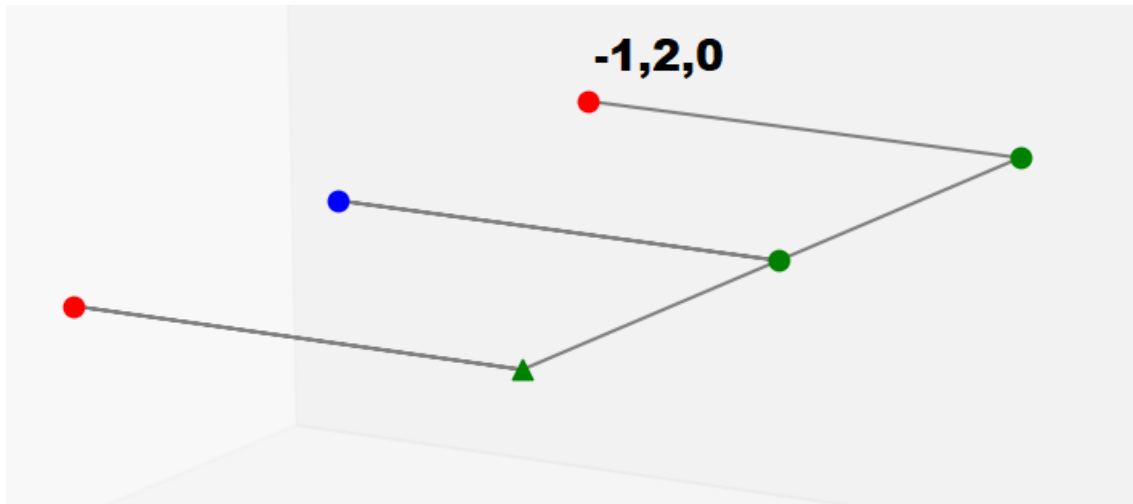


Figura 25: Orientação da terceira Cadeia Lateral relativa ao terceiro *backbone*

Fonte: Autoria Própria.

Neste trabalho os seguintes parâmetros foram utilizados na implementação do NSGA-II:

- a matriz de pesos de interações para o cálculo da energia livre (tabela 3);
- as distâncias entre pontos na treliça, padronizados como unitários;
- a probabilidade de mutação foi de 8% ;
- a seleção dos indivíduos foi realizada por torneio, onde 2 rodadas cada uma com 10 pais selecionados aleatoriamente onde os melhores de cada rodada foram selecionados de acordo com os critérios de dominância do NSGA-II em f_1 e f_2 .
- a quantidade de gerações foi mantida em 3000.
- o cruzamento de dois pontos foi o escolhido neste trabalho, neste tipo de cruzamento selecionam-se 2 pontos de maneira aleatória nos indivíduos a serem cruzados, dividindo os mesmos em 3 pontos. Os filhos gerados são resultado das do cruzamento do material genético entre os pontos selecionados. A figura 26 exemplifica este tipo de cruzamento e geração dos novos indivíduos;

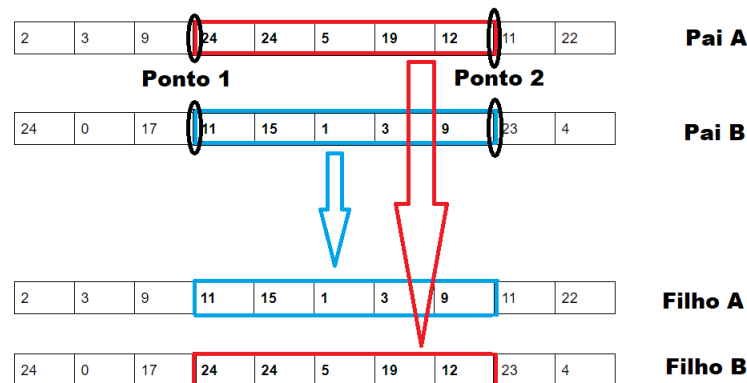


Figura 26: Cruzamento de dois pontos

Fonte: Autoria Própria.

- o gerador de número aleatórios para este trabalho foi o Mersenne Twister (Matsumoto e Nishimura 1998).
- a tabela 3 apresenta as ponderações de energia utilizadas.

Tabela 3: Pesos para as interações do sistema utilizado (Benitez e Lopes 2010)

Contato ϵ **	H	P	BB
H	10	-3	-3
P	-3	1	1
BB	-3	1	1

A seleção dos melhores indivíduos foi realizada utilizando 2 critérios sendo eles uma ordem de prioridades para escolha das melhores estruturas. Sendo eles:

- 1 - Indivíduos com menor energia livre f_2
- 2 - Indivíduos com mais contatos hidrofóbicos f_1

A ordem dos itens 1 e 2 foi criada visando a menor energia livre em um sistema. Caso a função f_1 tivesse prioridade sobre a função f_2 algumas estruturas poderiam ter energias menores com mais contatos hidrofóbicos, pois a equação de energia H e os pesos da tabela 3 permitem o surgimento dessas estruturas. Logo estruturas candidatas a melhor conformação (menor energia livre) poderiam ser preteridas por estruturas com o maior número de contatos hidrofóbicos.

É importante salientar que nas primeiras gerações muitas estruturas possuem colisões, porém as mais promissoras são mantidas de acordo com a função f_2 , pois uma colisão tende a ser penalizada de acordo com a equação 4.

O algoritmo 6 descreve o processo referente a verificação e contagem de colisões de estruturas: onde,

Algoritmo 6: Algoritmo de avaliação de colisões

Entrada: $Matriz_{x,y,z}$
Saída: *contador*

```

1 início
2   contador = 0
3   para Cada posição  $i$  da  $Matriz_{x,y,z}$  até a penúltima posição faça
4      $j = i + 1$ 
5     para Cada posição  $j$  de  $Matriz_{x,y,z}$  até a última posição faça
6       se coordenadas de  $Matriz_{x,y,z}[i]$  colidem com  $Matriz_{x,y,z}[j]$  então
7         contador = contador + 1
8       fim
9     fim
10  fim
11 fim
  
```

- $Matriz_{x,y,z}$ possui as coordenadas relativas de um indivíduo da população.
- *contador* realiza a contagem de quantos pontos da estrutura montada colidem.

O algoritmo 6 é parte intrínseca no cálculo de f_2 , pois ele define o número de colisões que irá penalizar o cada indivíduo. O tamanho de cada indivíduo é calculado pelo tamanho do indivíduo é calculado da seguinte maneira:

$$tamanho_{indiv\u00edduo} = (tamanho_{sequencia} - 1) * 2 \quad (15)$$

O decremento de 1 no tamanho da sequência se deve devido ao primeiro par BB-SC já ter suas posições estipuladas no espaço tridimensional.

O Algoritmo 7 é proposto com o intuito de conciliar as operações do algoritmo 4 do NSGA-II com os cálculos necessários para os eventos de dobra de proteína e o cálculo das funções f_1 e f_2 descritas anteriormente.

Algoritmo 7: Pseudo código do Algoritmo NSGA-II 3DHPSC

Entrada: Parâmetros do NSGA II

Saída: Soluções $P_{N_{iteracoes}} \cup Q_{N_{iteracoes}}$

```

1 início
2   Lê sequência  $Seq$  de um arquivo de entrada
3   Calcula o tamanho de cada indivíduo( $2 * Sequencia$ )
4   Criar população inicial aleatória  $P_1$  de  $N_{individuos}$  linhas por  $(seq-1)$  colunas
5   Preenche cada indivíduo de  $P_1$  com números aleatórios entre 0 e 24
6   Aplica o algoritmo 5 para cada indivíduo de  $P_1$ 
7   Calcula colisões de acordo com o algoritmo 6 para cada indivíduo
8   Calcula  $f_1$  e  $f_2$  para cada indivíduo de acordo com a  $matriz_{x,y,z}$  e as colisões
   de cada indivíduo
9   Ordenar  $P_1$  por dominância de acordo com  $f_1$  e  $f_2$ 
10  Aplicar operadores genéticos em  $P_1$  para gerar uma nova população  $Q_1$  de
   tamanho  $N_{individuos}$ 
11  para Geração  $t = 1, 2, \dots$  até  $t = N_{iteracoes}$  faça
12    Aplicar Ordenação por dominância em  $P_t$  e  $Q_t$  e gera as Fronteiras
13     $k = 1$ 
14    repita
15      Realizar calculo da Distância de Multidão em  $F_k$ 
16       $P_{t+1} = P_{t+1} \cup F_k$ 
17      incrementa  $k$  em 1;
18    até  $P_{t+1} + F_k > N_{individuos}$ ;
19    Gerar nova população  $Q_{t+1}$  aplicando os operadores genéticos
20    Aplica o algoritmo 5 para cada indivíduo de  $P_{t+1}$  e  $Q_{t+1}$ 
21    Calcula colisões em  $Q_{t+1}$  e  $P_{t+1}$  de acordo com o algoritmo 6 para cada
   indivíduo
22    Calcula  $f_1$  e  $f_2$  para cada indivíduo de acordo com a  $matriz_{x,y,z}$  e as
   colisões de cada indivíduo
23  fim
24  retorna  $P_{N_{iteracoes}} \cup Q_{N_{iteracoes}}$ 
25 fim

```

No trabalho 2 computadores de configuração i7-4930K de 6 núcleos e 24 Gigabytes de memória foram utilizados e o algoritmo foi implementado de forma sequencial na linguagem python 3.4. Nenhuma biblioteca para ganho de desempenho ou paralelização foi utilizada. Alguns testes foram realizados com a utilização da biblioteca Numba¹² em algumas funções do algoritmo, como as de montagem tridimensional e os cálculos de colisões, mas durante os primeiros experimentos notou-se que não alterava o tempo de execução do algoritmo, logo optou-se por não se fazer utilização da mesma.

No trabalho de (Benítez 2010) estratégias de paralelização foram utilizadas, logo o tempo computacional não será comparado. As duas funções objetivo propostas serão apresentadas porém, como o trabalho comparativo utiliza uma abordagem mono-objetivo será comparada apenas a quantidade de contatos hidrofóbicos, porém a energia livre obtida neste trabalho estará disponível.

O fato de se utilizar uma esquema sequencial indicou que a população utilizada nos experimentos deveria ser menor que a utilizada no experimento comparado (foi de 500 indivíduos). Foram testadas populações de 75, 100, 200, 300 e 500. As populações de 75 e 100 foram descartadas durante a avaliação das primeiras execuções do algoritmo pois apresentaram poucas variações nas soluções recebidas. As populações de 300 e 500 apesar de obterem resultados interessantes nas primeiras execuções foram descartadas pois levaram muito tempo sendo executadas. Logo, a população que empiricamente se adequou a este trabalho foi de 200 indivíduos.

O critério de parada ficou estipulado em 3000 gerações, pois não há outra informação sobre a melhor estrutura a ser obtida pelo algoritmo além da sequencia de aminoácidos. Durante o processo de calibração dos parâmetros do algoritmo foram testadas estratégias de dizimação de população onde tentou-se eliminar os piores indivíduos e troca-los por novos indivíduos gerados aleatoriamente. A estratégia de dizimação foi testada com 50, 100 e 200 gerações de estagnação e notou-se que não alteraria o resultado final, pois os novos indivíduos gerados seriam eliminados por causa processo de torneio, que seleciona os melhores indivíduos, e em caso de cruzamento entre indivíduos gerados no processo de classificação das fronteiras eles eram eliminados pois não conseguiam ser melhores que os seus pais ou os filhos gerados por pais mais aptos, ficando em fronteiras dominadas por muitas fronteiras.

No processo de dizimação também foram testadas as quantidades de indivíduos a serem dizimados, entre 25, 50 e 100 indivíduos, consistindo na eliminação dos indivíduos com

¹<https://numba.pydata.org/>

²Biblioteca do python que transforma trechos do código python em linguagem de maquina, com a o objetivo de tornar sua execução mais rápida, possui a opção de paralelização dos núcleos do processador

menor aptidão após a formação de uma nova geração, sendo que independente da quantidade, os resultados não foram promissores pois não alteraram a estagnação da execução do algoritmo. Sendo assim, este tipo de estratégia foi descartada pois não foi eficiente na produção de novos indivíduos e foi computacionalmente custosa, aumentando o tempo de execução do algoritmo.

Os *datasets* utilizados foram "dill.*" (Yue e Dill 1993), "Unger273.d.*" (Unger e Moulton 1993) e "S48.*" (Yue et al. 1995). A tabela 4 apresenta as sequencias utilizadas, n indica o número de aminoácidos da sequencia, H_{max} o numero de contatos hidrofóbicos máximos.

Tabela 4: Sequências de Benchmark para o modelo 3DHPSC

Referência	n	Sequencia	H_{max}
Dill.1	27	$HP^4H^4P(PH)^3H(HP)^2PH^2P^2H$	21
Dill.2	27	$HP^3H^4(PH)^2HP^3HPH(HP)^2P^2HP$	19
Dill.3	27	$HPH^2(PPHH)^2H(HPPP)^2H^3P^2H$	23
Dill.4	31	$(HHP)^3H(HHHHHPP)^2H^7$	41
Dill.5	36	$PH(PPH)^{11}P$	14
Unger273d.1	27	$(PH)^3H^2P^2(HP)^2P^{10}H^2P$	12
Unger273d.2	27	$PH^2P^{10}H^2P^2H^2P^2HP^2HPH$	13
Unger273d.3	27	$H^4P^5HP^5H^3P^8H$	13
Unger273d.4	27	$H^3P^2H^4P^3(HP)^2PH^2P^2HP^3H^2$	22
Unger273d.5	27	$H^4P^4HPH^2P^3H^2P^{10}$	13
Unger273d.6	27	$HP^6HPH^3P^2H^2P^3HP^4HPH$	14
Unger273d.7	27	$HP^2HPH^2P^3HP^5HPH^2(PH)^3H$	16
Unger273d.8	27	$HP^{11}(HP)^2P^7HPH^2$	6
Unger273d.9	27	$P^7H^3P^3HPH^2P^3HP^2HP^3$	10
Unger273d.10	27	$P^5H(HP)^5(PHH)^2PHP^3$	14
S48.1	48	$HPH^2P^2H^4PH^3P^2H^2HPH^3(PH)^2HP^2P^3HP^8H^2$	32
S48.2	48	$H^4(PHH)^2H^3(PPH)^2HP^2HP^6(HPP)^2PHP^2H^2P^2H^3PH$	32
S48.3	48	$(PH)^2HP^6P^2(HP)^2(PH)^2(HP)^3(PPH)^2HP^2H^2P^2(HP)^2PHP$	30
S48.4	48	$(PH)^2HP^2HPH^3P^2H^2PH^2P^3H^5P^2HPH^2(PH)^2P^4HP^2(HP)^2$	35
S48.5	48	$P^2HP^3HPH^4P^2H^4(PHH)^2HP(PH)^3P^2HP^5(PHH)^2PH$	31
S48.6	48	$H^3P^3H(PH)^2(HHP)^3HP^7(HP)^2PHP^3HP^2H^6PH$	33
S48.7	48	$PHP^4HPH^3(PH)^2H^3(PHH)^2P^3(HP)^2P^2H^3(PPHH)^2P^3H$	32
S48.8	48	$(PHH)^2HPH^4P^2H^3P^6HPH^2P^2H(HP)^2H^2(PH^3)HP^3$	31
S48.9	48	$(PH)^2P^4(HP)^3(PH)^2H^5P^2H^3PHP(PH)^2HP(PH)^2HP(PH)^2H^2P^4H$	32
S48.10	48	$PH^2P^6H^2P^3PHP(PH)^2(HPP)^3H^2P^2H^7P^2H^2$	33

4 EXPERIMENTOS E DISCUSSÃO

Neste capítulo serão apresentados os resultados de cada experimento de acordo com o grupo de *benchmark* ao qual pertence. Todas as sequencias de cada *dataset* foi executada 30 vezes, conforme sugere (Zitzler et al. 2001) em seus experimentos de abordagens multiobjetivo.

4.1 PARÂMETROS

Os parâmetros utilizados na implementação do algoritmo NSGA-II 3DHPSC foram os seguintes :

- Para o fator de mutação foi mantida a probabilidade de 8%;
- Para o vetor de movimentos de 5 bits foi utilizada uma alteração dos 2 bits mais significantes para a codificação se manter entre valores de 0 e 24 Sendo Assim, uma probabilidade de 50% de alteração dos valores dos bits mais significantes (2 primeiros bits a esquerda), sendo que no mínimo um bit é selecionado para ser alterado. Desta forma é garantida a margem de 25 movimentos em valores de 0 a 24, a figura 27 representa essa alteração, onde um vetor binário contém o valor 25 decimal que não compreende a faixa de valores permitida, ele sofre a alteração de 1 dos 2 primeiros bits a esquerda para adequação da faixa de valores, chegando ao numero 9 decimal, que se adequá aos movimentos permitidos neste estudo ;
- População inicial de 200 indivíduos, observando-se que os filhos são gerados de cruzamentos aleatórios entre os 200 indivíduos, formando uma nova população de tamanho 200;
- Para o $Vetor_{ref}$ a seguinte orientação foi utilizada para nortear a próxima estrutura *backbone-side chain*:
- o cruzamento selecionado foi o de dois pontos.

F	C	B	E	D	Lados	Frente	Cabeça-Pé
1	1	-1	-1	1	2	0	1

- A geração máxima é 3000, sendo o único critério de parada do algoritmo.
- O algoritmo de aleatoriedade foi o Mersenne Twister (Matsumoto e Nishimura 1998).
- Para classificar as fronteiras o método KNN foi implementado no calculo da distância de multidão, onde os vizinhos mais próximos classificam a qual fronteira o indivíduo pertence, sendo a primeira fronteira a que define as demais, pois não é dominada por nenhum vizinho.

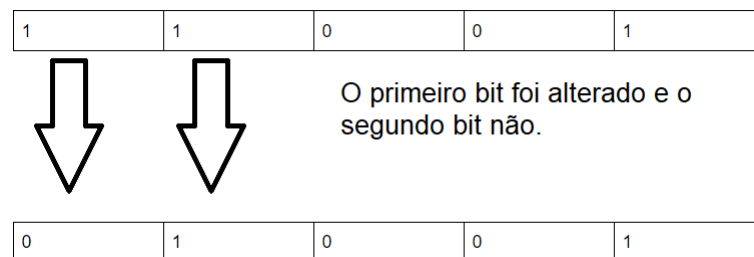


Figura 27: Exemplo da alteração do bit mais significativo do vetor de movimentos

Fonte: Autoria Própria

4.2 DISCUSSÃO DOS RESULTADOS

Para análise das tabelas:

- Na coluna referência estão os *datasets* correspondentes aos resultados obtidos;
- Na coluna geração estão os melhores resultados de cada *dataset* seguido da média dos 30 experimentos e o desvio padrão;
- Na coluna T_s estão os tempos de execução médios em segundos dos 30 experimentos;
- Na coluna HNC estão as médias de todos os 30 experimentos em relação aos contatos hidrofóbicos seguidos do desvio padrão e o melhor resultado obtido obedecendo os 3 critérios para a seleção da melhor estrutura.
- Na coluna f_2 estão as médias das energias livres seguidas do desvio padrão e o melhor resultado obtido obedecendo os 3 critérios para a seleção da melhor estrutura.
- A coluna HNC_{ref} representa os resultados obtidos no trabalho de (Benítez 2010) com a média dos experimentos realizados em relação aos contatos hidrofóbicos seguidos do desvio padrão e o melhor resultado obtido para cada experimento do *dataset*.

Tabela 5: Resultados para o *dataset* Dill.*

Referência	Geração		T_s	HNC		f_2		HNC_{ref}	
	Melhor	Média		Média	Melhor	Média	Melhor	Média	Melhor
Dill.1	1565	2326,25 ± 590,00	23510	12,72 ± 1,88	16	- 595,87 ± 125,30	- 829,00	17,42 ± 1,86	21
Dill.2	1763	2060,75 ± 803,03	24997	9,37 ± 1,31	12	- 461,37 ± 58,44	- 545,40	14,78 ± 1,56	19
Dill.3	2886	2145,26 ± 826,77	25651	12,63 ± 1,68	17	- 717,91 ± 142,68	- 1016,83	18,00 ± 1,79	23
Dill.4	2593	2026,11 ± 732,46	28447	21,25 ± 2,85	30	- 1280,41 ± 234,35	- 1999,00	32,81 ± 2,40	41
Dill.5	1497	2002,07 ± 907,35	39366	7,82 ± 1,46	11	- 589,08 ± 115,65	- 824,53	11,29 ± 1,27	14

Ao analisar a tabela 5 e realizar um comparativo simples nos contatos hidrofóbicos obtidos entre os experimentos deste trabalho (HNC) e os de (Benítez 2010) (HNC_{ref}) é possível notar que os experimentos obtiveram resultados inferiores aos obtidos nos experimentos mono-objetivo propostos anteriormente, pois nenhum resultado conseguiu obter um número de contatos hidrofóbicos melhor.

Em média, o algoritmo estagna perto das 2000 gerações para este *dataset*, independente do tamanho da sequência. As sequências com mais contatos hidrofóbicos Dill.3 e Dill.4 estagnaram após 2500 gerações, enquanto que o pior desempenho foi da sequência Dill.5 (36 resíduos), pois possui a pior média de de variação de gerações e o estagnou com menos de metade de gerações restantes.

Tabela 6: Resultados para o *dataset* Unger273d.*

Referência	Geração		T_s	HNC		f_2		HNC_{ref}	
	Melhor	Média		Média	Melhor	Média	Melhor	Média	Melhor
Unger273d.1	948	2305,75 ± 661,23	25814	6,97±0,91	8	- 462,13± 81,46	- 600,34	10,06 ± 1,18	12
Unger273d.2	2694	2029,28 ± 623,45	24554	7,71 ± 1,17	10	- 528 ± 104,89	- 751,15	11,89 ± 0,93	13
Unger273d.3	2419	1967,31 ± 848,90	26165	7,30 ± 1,44	11	- 553,48 ± 139,98	- 902,21	10,71 ± 0,94	13
Unger273d.4	2913	1945,39 ± 941,19	25466	11,06±1,53	14	- 644,01±105,15	- 838,98	17,79 ± 2,20	22
Unger273d.5	1914	2057,69 ± 775,40	24185	7,66 ± 1,05	9	- 514,91 ± 66,08	- 620,3	10,83 ± 1,29	13
Unger273d.6	1852	1905,78±883,46	25297	7,19±1,30	10	- 387,52±75,36	- 567,32	11,28 ± 1,29	14
Unger273d.7	1647	1951,68±726,66	24029	7,93±1,43	11	- 521,86±102,78	- 781,17	12,57± 1,50	16
Unger273d.8	2070	2062,33±869,43	24396	3,03±0,93	5	- 349,73±-99,25	- 555,48	4,68 ± 0,85	6
Unger273d.9	2542	1965,62±853,29	24377	6,34±0,98	8	- 256,00±45,05	- 348,04	8,06 ± 1,03	10
Unger273d.10	2159	1967,31±848,90	26695	7,96±1,40	11	- 306,27± 64,52	- 437,44	11,08 ± 1,23	14

Ao se analisar os resultados dos experimentos para o *dataset* Unger273d.* é possível notar que o *HNC* deste estudo se aproximou dos obtidos na referência HNC_{ref} , com exceção do experimento Unger273.4 que foi muito inferior. As execuções tenderam a estabilizar próximas a 2000 gerações porém, 4 dos 10 melhores resultados estabilizaram com menos.

É importante salientar que para este *dataset* as sequências possuem tamanhos iguais a 27 resíduos. Apesar dos melhores indivíduos serem semelhantes em quantidades de contatos hidrofóbicos, se comparada a média dos experimentos, a discrepância entre o desempenho com relação a busca de melhores resultados fica evidente.

Tabela 7: Resultados para o *dataset* S48.*

Referência	Geração		T_s	HNC		f_2		HNC_{ref}	
	Melhor	Média		Média	Melhor	Média	Melhor	Média	Melhor
S48.1	2963	2450,88±547,94	64795	16,33±2,16	17	- 1495,37±226,99	- 1787,50	26,24± 3,13	32
S48.2	2911	2472,10±586,31	57424	16,16±1,65	18	- 1766,21±346,53	- 2402,07	26,68 ± 3,07	32
S48.3	2788	2328,54±599,72	64367	14,72±2,02	17	- 1403,73±256,14	- 1956,86	24,63 ± 3,32	30
S48.4	2768	2401,50±596,13	57528	14,90±2,41	18	- 1333,99±229,97	- 1810,23	25,75 ± 3,47	35
S48.5	2064	2273,32±671,44	62808	15,54±1,99	17	- 1391,02±229,13	- 1839,17	26,37 ± 2,87	31
S48.6	2943	2356,12±680,90	57381	15,57±2,99	21	- 1708,82±390,35	- 2593,75	26,29 ± 3,04	33
S48.7	2432	2236,78±751,77	62646	15,87±2,21	21	- 1351,18±-276,05	- 1934,55	24,63 ± 3,56	32
S48.8	2747	2243,06±639,13	56494	15,57±2,13	21	- 1517,55±293,65	- 2078,59	25,94 ± 2,89	31
S48.9	2708	2416,68±575,66	65132	15,87±1,78	18	- 1301,22±144,30	- 1685,61	26,12 ± 2,99	32
S48.10	2424	2443,00 ± 540,77	57113	17,33±2,40	24	- 1765,85±330,81	- 2548,87	27,75 ± 2,89	33

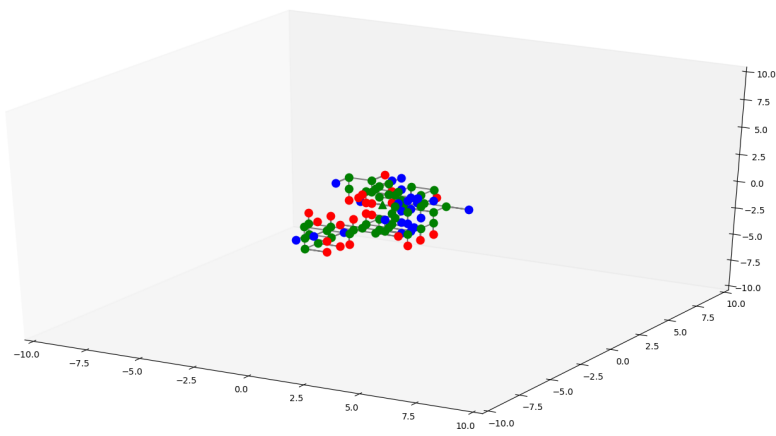
Os resultados do *dataset* S48.* da tabela 7 foram os piores se comparados com os obtidos por (Benítez 2010), apesar de serem os que mais variaram conforme a execução dos experimentos. Essa variação ocorreu por conta do tamanho do sequencia ser maior (48 resíduos), o que torna o espaço de busca maior e por consequência, o algoritmo é mais suscetível a encontrar melhorias nas estruturas encontradas. Porém a baixa variação de contatos hidrofóbicos indica que o algoritmo busca soluções em ótimos locais, variando pouco no núcleo hidrofóbico.

Os tempos de execução médios foram superiores ao dobro dos experimentos *dataset* Unger273d.*, mesmo o tamanho das sequências não dobrando, isto se deve a natureza exponencial do problema de dobramento de proteínas.

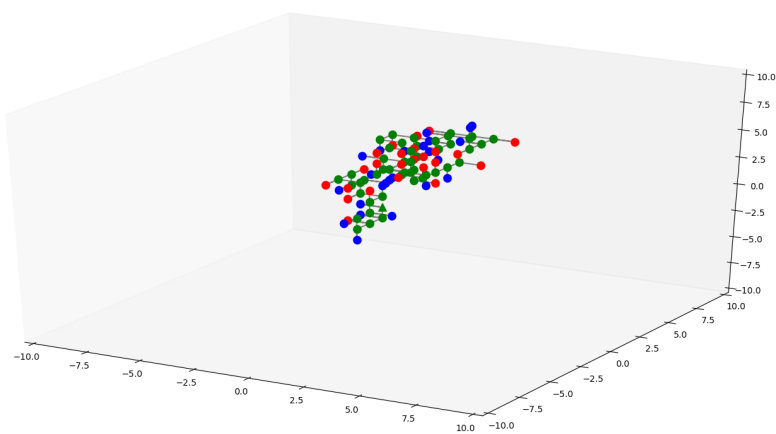
Algumas estruturas obtiveram *HNC* maiores do que os apresentados na nas tabelas 5, 6 e 7 porém com F_2 inferiores, a escolha pela melhor estrutura se deu devido a preferência por uma energia livre F_2 menor. Para exemplificar, utilizaremos a estrutura gerada pela sequência S48.5 do *dataset* S48.*, onde foi possível obter uma estrutura válida com mais contatos hidrofóbicos, porém com energia livre maior que a energia livre do resultado presente na tabela 7.

A figura 28 apresenta a estrutura presente na tabela 7 e considerada melhor de acordo com a menor energia livre(a) e a estrutura com mais contatos hidrofóbicos referida(b).

As esferas vermelhas correspondem às estruturas hidrofóbicas, as esferas azuis às estruturas polares e as esferas verdes são os *backbones*. O triângulo verde é o *backbone* inicial na referência do plano tridimensional(0,0,0).



(a) Estrutura S48.5a



(b) Estrutura S48.5b

Figura 28: Estruturas S48.5 com o mesmo número de contatos hidrofóbicos, mas com energia livre diferentes

Observando a figura 28 é possível verificar que a estrutura S48.5a obteve maior compactação que a estrutura S48.5b gerada, mesmo possuindo menor número de contatos hidrofóbicos, o que será apresentado matematicamente mais adiante. A escolha se deu obedecendo os critérios descritos no capítulo 3 e as equações 3 e 4 descritas na seção 3.5.1.1 onde os contatos seguem conforme a tabela 8 a seguir:

Tabela 8: Contatos por experimento no caso S48.5

Contatos	HH	HP	HB	PB	PP	BB
S48.5a	17	3	4	14	9	6
S48.5b	20	8	6	15	3	6

Sendo,

- HH os contatos hidrofóbico-hidrofóbico;
- HP os contatos hidrofóbico-polar;
- HB os contatos hidrofóbico-backbone;
- PB os contatos polar-backbone;
- PP os contatos polar-polar;
- BB os contatos backbone-backbone.

Aplicando-se os contatos da tabela 8 e os pesos das interações descritos na tabela 9 na equação 16 obtemos os valores de energia para S48.5a de 178 e de S48.5b de 182.

Tabela 9: Pesos para as interações do sistema utilizado (Benitez e Lopes 2010)

Contato ε^{**}	H	P	BB
H	10	-3	-3
P	-3	1	1
BB	-3	1	1

$$\begin{aligned}
 H = & \varepsilon_{HH} \sum_{i=1, j>1}^N \delta_{r_{ij}^{HH}, a} + \varepsilon_{BB} \sum_{i=1, j>i+1}^N \delta_{r_{ij}^{BB}, a} + \\
 & \varepsilon_{BH} \sum_{i=1, j \neq i}^N \delta_{r_{ij}^{BH}, a} + \varepsilon_{BP} \sum_{i=1, j \neq i}^N \delta_{r_{ij}^{BP}, a} + \\
 & \varepsilon_{HP} \sum_{i=1, j>i}^N \delta_{r_{ij}^{HP}, a} + \varepsilon_{PP} \sum_{i=1, j>i}^N \delta_{r_{ij}^{PP}, a}
 \end{aligned} \tag{16}$$

Em ambos os casos o operador *RadiusGP* foi igual a 1. Operador *RadiusGh* para S48.5a foi de 10,3324 e para S48.5b foi 10,0171. Estes valores indicam que a estrutura S48.a é

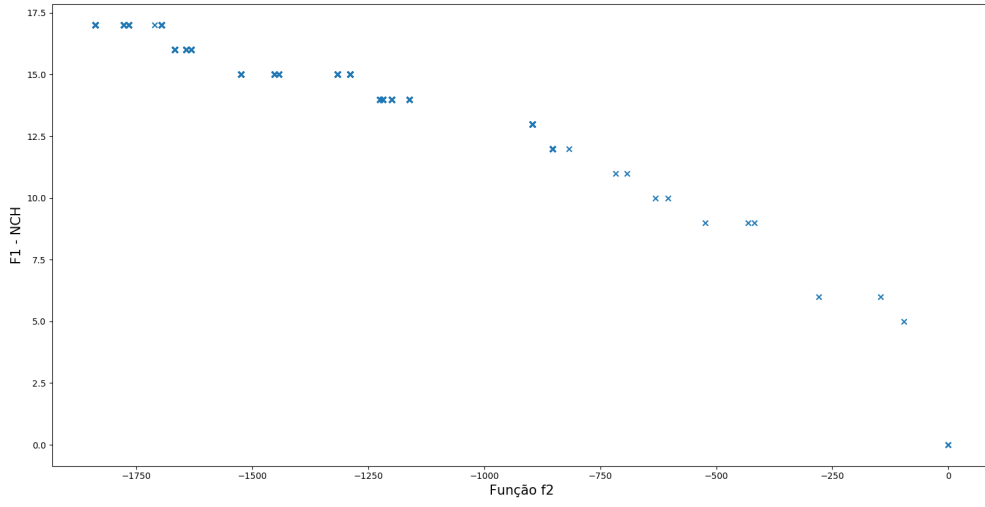
a mais compacta dentre as duas pois quanto maior o termo *RadiusGh*, mais compacta a estrutura está, sabendo que o raio de giração máximo (*MaxRGH*) para este experimento é 13,1529.

$$f_2 = -(Energia * RadiusGh * RadiusGP) \quad (17)$$

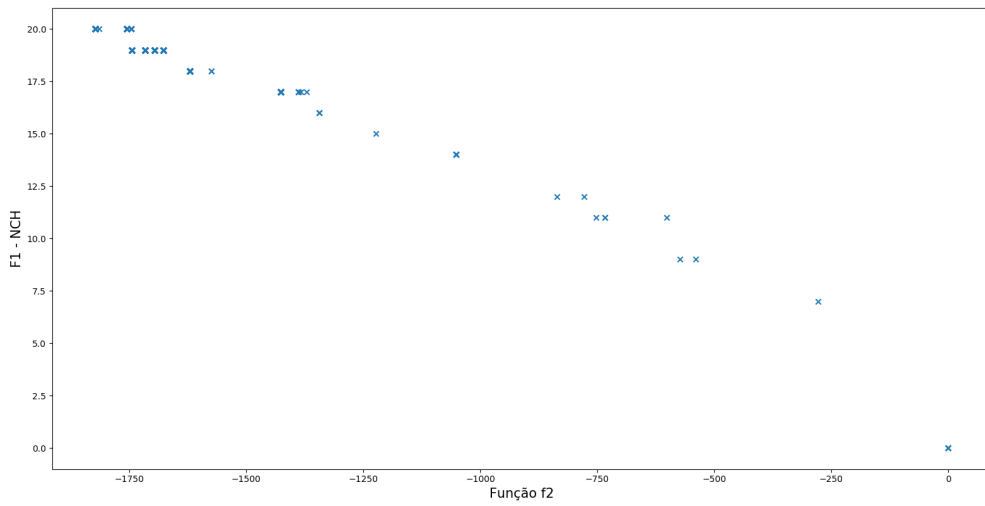
Sendo assim a energia f_2 descrita na equação 17 para S48.5a é igual a $-1839,1784$ e para S48.5b é igual a $-1823,1132$.

Estes resultados indicam que a estrutura S48.5a é a melhor estrutura pois obtém a menor energia livre F_2 para o sistema, apesar de obter menos contatos, além de a estrutura tridimensional apresentar uma tendencia das estruturas hidrofóbicas estarem centralizadas na estrutura.

Ainda sobre a análise dos experimentos S48.a e S48.5b figura 29 apresenta a distribuição do crescimento do número de contatos hidrofóbicos *HNC* em relação a diminuição da energia livre da estrutura f_2 .



(a) HNC X F2 S48.5a

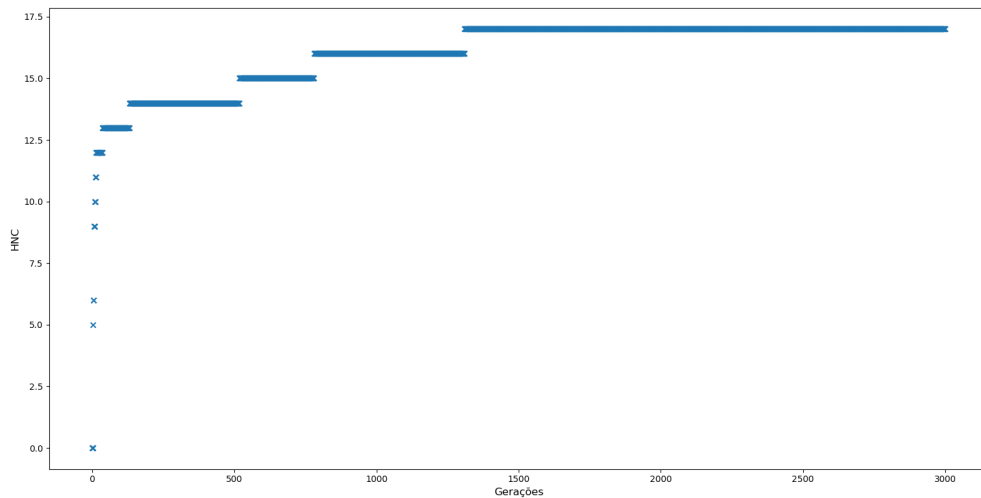


(b) HNC X F2 S48.5b

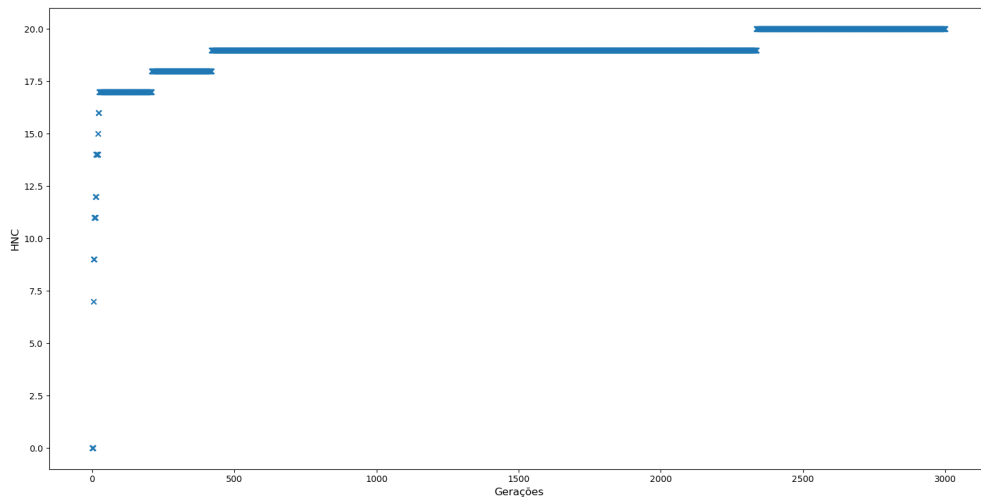
Figura 29: HNC X F2 S48.5

A figura 30 apresenta o crescimento dos contatos hidrofóbicos para S48.5a e S48.5b de acordo com o avanço das gerações e a figura 31 apresenta a variação da energia livre f_2 em relação as gerações.

Ao analisar o comportamento dos gráficos 30 (a) e (b) é possível notar as tendencias de estabilização do algoritmo, pois a partir de certo ponto (aproximadamente 1500 para (a) e 500 para (b)) poucas variações no número de contatos ocorrem.



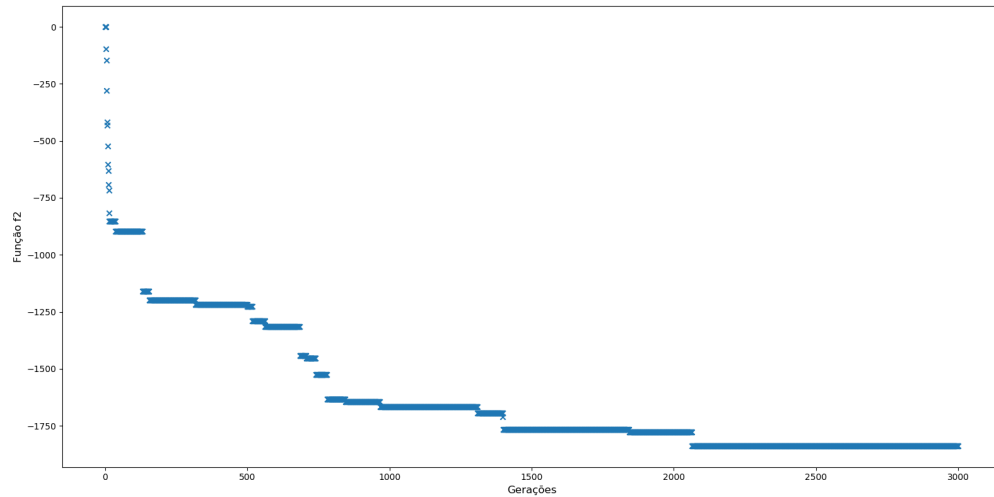
(a) HNC S48.5a



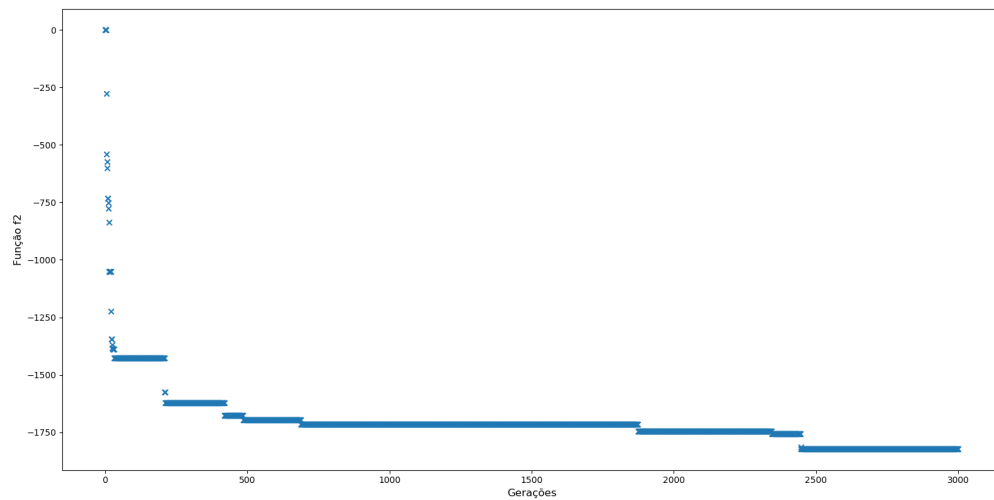
(b) HNC S48.5b

Figura 30: HNC S48.5

Com a análise do 30 e a análise e o comportamento dos gráficos (a) e (b) das figura 31 é possível notar que a energia livre F_2 continuou diminuindo com o passar das gerações, porém com poucas mudanças nos contatos hidrofóbicos HNC , ou seja, ilhas de contatos são formadas e apenas algumas estruturas polares e *backbones* são movimentadas, fazendo pouco efeito sobre a compactação da estrutura ou a formação do núcleo hidrofóbico.



(a) F2S48.5a

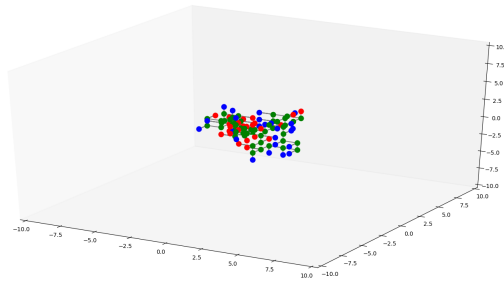


(b) F2 S48.5b

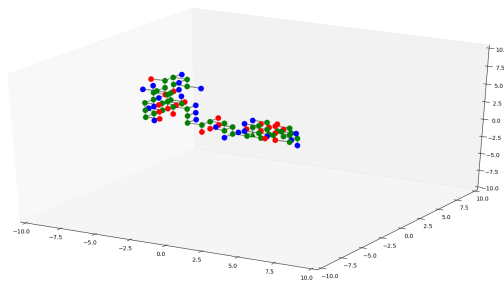
Figura 31: F2 S48.5

Quando analisados os gráficos dos experimentos (em sua totalidade no Apêndice A deste trabalho) é possível concluir que o algoritmo tende a criar ilhas de contatos hidrofóbicos estagnado em ótimos locais. Para exemplificar a figura 32 apresenta as estruturas tridimensionais dos experimentos S48.6, S48.7 e S48.8 que obtiveram o mesmo número de contatos hidrofóbicos, porém disposições e sequências diferentes. Para o caso específico do experimento S48.7, a menor quantidade de aminoácidos hidrofóbicos e a distancia entre eles colaborou para formação das ilhas nas extremidades.

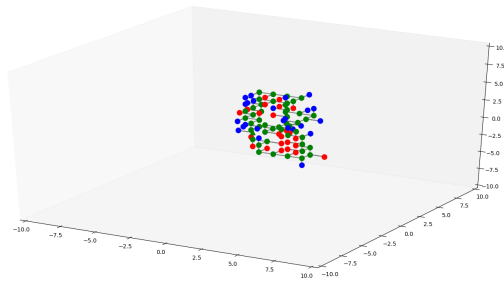
Essa estagnação das estruturas e o fato de as variações serem apenas de estruturas polares ou *backbones* após um certo número de gerações sugere que o NSGA-II implementado neste estudo é ineficiente para resolver o problema combinatório do modelo 3DHPSC, pois ao encontrar um ótimo local ele não consegue explorar outras regiões com soluções mais promissoras que as encontradas.



(a) Estrutura S48.6



(b) Estrutura S48.7



(c) Estrutura S48.8

Figura 32: Estruturas S48.6 , S48.7 e S48.8

5 CONSIDERAÇÕES FINAIS

O problema de dobramento de proteínas vem sendo explorado pelos diversos campos das ciências, do ponto de vista computacional se torna um desafio pois até o modelo de dobramento mais simples é um problema NP-Difícil. Como consequência, estudos utilizando aprendizado de máquina e métodos heurísticos vem sendo propostos com o intuito de ajudar a resolver o PDP e mesmo aplicados a modelos mais simples, esses métodos contribuem para que novos algoritmos de otimização surjam, podendo posteriormente serem utilizados em modelos complexos e por consequência obterem resultados que possam auxiliar no melhor entendimento das proteínas e o seu processo de dobra.

O modelo 3DHPSC tem sido pouco explorado na literatura, em comparação com seu modelo homólogo o 3DHP, devido a sua complexidade. Logo este estudo apresenta contribuições importantes para abordar este modelo propondo uma abertura da função objetivo apresentada (Li et al. 2002) e explorada por (Benítez 2010), que foi decomposta para formular um problema biobjetivo, em busca de encontrar novas soluções para o modelo.

O algoritmo NSGA-II foi escolhido pois é um algoritmo de otimização multiobjetivo vastamente na literatura para resolver os mais variados problemas de otimização, que por se basear em GAs ajudou a utilizar alguns dos parâmetros presentes no trabalho de (Benítez 2010). Foi possível implementar o algoritmo NSGA-II 3DHPSC para realizar os testes com as sequencias *benchmark* e comparar com os resultados obtidos por (Benítez 2010), sendo assim os objetivos desse estudo foram alcançados.

Algumas alterações com relação ao estudo proposto anteriormente por (Benítez 2010) foram necessárias por limitações de hardware, porém os 25 experimentos foram executados as 30 vezes recomendadas pela literatura, o que demandou de muitos dias para a execução. É importante salientar que um dos fatores que fizeram os resultados deste trabalho não alcançarem os resultados de (Benítez 2010) se deve pelo uso de uma população menor (200 ao invés de 500) o que prejudicou a diversidade da população gerada, e por consequência o alcance das possibilidades geradas pelos operadores do NSGA-II 3DHPSC implementado. O tempo

de execução também foi superior pelo fato do algoritmo implementado neste trabalho ser sequencial e o utilizado no trabalho comparado utilizou técnicas de paralelismo para execução.

Apesar dos resultados serem inferiores aos da literatura, alguns pontos podem ser ressaltados, como a quantidade de gerações que tende a estabilizar com estruturas mais simples por volta das 2000 gerações, o que indica que uma população maior pode convergir para resultados melhores com menos gerações, sem a necessidade de alterar os operadores.

Sobre o fator de mutação do algoritmo, nenhuma estratégia foi proposta pois como descrito anteriormente, o algoritmo consumia muito tempo para computar as 3000 gerações, e no trabalho de (Benítez 2010), é realizado um estudo sobre o operador de mutação, onde por exemplo, se fosse utilizada uma comparação entre o indivíduo antes e depois da mutação para selecionar qual a melhor estrutura, o algoritmo se tornaria consideravelmente mais custoso, o que indicou que este tipo de estudo seria inviável para este trabalho dadas as limitações de hardware.

Outro ponto importante a ser salientado é que a própria montagem e manipulação da estrutura da proteína é custosa computacionalmente, e a inclusão da cadeia lateral aumenta consideravelmente este custo, então um próximo passo importante para o estudo realizado neste trabalho, é o planejamento de novas estratégias para a montagem da estrutura para os cálculos a serem realizados. Diferente de (Benítez 2010) que utilizou coordenadas relativas para montagem e orientação das estruturas, este trabalho utilizou a estratégia de coordenadas absolutas, o que pode ser reavaliado em estudos futuros.

Uma alternativa que surge em tempos atuais tanto para implementação de novas estratégias e otimização dos parâmetros é a utilização de paralelização do algoritmo com a utilização de GPUs, que tem se mostrado uma ótima solução para execução de cálculos computacionais. Com a paralelização, poderá ser possível uma análise mais precisa do algoritmo visto que, comparativos poderam ser realizados com os tempos obtidos nos experimentos mono-objetivo de (Benítez 2010).

Uma implementação do algoritmo SPEA-II foi realizada após os resultados do NSGA-II serem obtidos, para o estudo ter ao menos mais um critério de comparação de métodos multiobjetivo, porém o algoritmo utilizando parâmetros semelhantes os do NSGA-II deste demorou cerca de 96 horas para concluir as 3000 gerações dos primeiros experimentos, concluiu-se que era inviável no momento a realização deste comparativo.

Para este projeto em estudos futuros pretende-se:

- A implementação estratégias de *backtracking* combinadas com buscas locais para

aproveitar as estruturas geradas ao invés de contar apenas com os operadores do NSGA-II. O *backtracking* também pode realizar rotações em sub-estruturas promissoras para gerar conformações melhores para as proteína. Estas buscas locais podem ser implementadas com a utilização de paralelismo, visto que um dos momentos mais críticos da execução deste algoritmo está na montagem tridimensional e calculo das interações entre os aminoácidos e as cadeias laterais.

- Realizar estudos de ajuste de operadores para o NSGA-II 3DHPSC, em busca de melhores resultados, exemplos deste tipo de estudo podem ser encontrados em (Maruo et al. 2005, Tsunoda e Constâncio).
- Realizar estudos sobre a codificação dos indivíduos, observando as formas de representação apresentadas por (Hart et al. 1999, Lopes 2008) e avaliando seu desempenho.
- Implementação de outros algoritmos multiobjetivo, como por exemplo o algoritmo SPEA-II para comparativos de desempenho e resultados dos algoritmos, visto que o NSGA-II e o SPEA-II exploram a Fronteira de Pareto de maneira diferente.
- Realizar uma implementação com maior eficiência que (Benítez 2010), com relação ao número de contatos e formação do núcleo hidrofóbico, para poder comparar com os resultados obtidos por (Hattori et al. 2020) em seu estudo recente com sequências do Protein Data Bank.
- Avaliação de novas estratégias de dizimação, visto que apenas a eliminação de indivíduos menos aptos não foi eficiente.

Este trabalho também conclui que apesar do NSGA-II ser um algoritmo altamente explorado na literatura, se mostrou pouco eficiente com o problema de dobramento de proteína utilizando o modelo 3DHPSC, pois seu foco são soluções não-dominadas e o elitismo das soluções acaba por eliminar estruturas promissoras por estarem em fronteiras que não estarão em novas gerações, assim sendo, ao menos na execução deste trabalho ele se mostrou ineficiente para sair de ótimos locais. Porém os estudos futuros propostos devem direcionar ele para soluções mais eficientes saindo da estagnação em ótimos locais.

REFERÊNCIAS

- ALBERTS, B. et al. *Molecular biology of the cell* (garland, new york, 1994). **Google Scholar**, p. 907–982.
- ALMEIDA, A. B. d. et al. Predição de estrutura terciária de proteínas com técnicas multiobjetivo no algoritmo de monte carlo. Universidade Federal de Goiás, 2016.
- ANFINSEN, C. B. Principles that govern the folding of protein chains. **Science**, JSTOR, v. 181, n. 4096, p. 223–230, 1973.
- BAKER, B. S.; BHATT, S. N.; LEIGHTON, F. T. An approximation algorithm for manhattan routing. In: **ACM. Proceedings of the fifteenth annual ACM symposium on Theory of computing**. [S.l.], 1983. p. 477–486.
- BENÍTEZ, C. M. V. **Um algoritmo genético paralelo para o problema de dobramento de proteínas utilizando o modelo 3DHP com cadeia lateral**. Dissertação (Mestrado) — Universidade Tecnológica Federal do Paraná, 2010.
- BENITEZ, C. M. V.; LOPES, H. S. Hierarchical parallel genetic algorithm applied to the three-dimensional hp side-chain protein folding problem. In: **IEEE. 2010 IEEE International Conference on Systems, Man and Cybernetics**. [S.l.], 2010. p. 2669–2676.
- BENÍTEZ, C. M. V.; LOPES, H. S. Parallel artificial bee colony algorithm approaches for protein structure prediction using the 3dhp-sc model. In: **Intelligent Distributed Computing IV**. [S.l.]: Springer, 2010. p. 255–264.
- BENÍTEZ, C. M. V.; LOPES, H. S. Molecular dynamics for simulating the protein folding process using the 3d ab off-lattice model. In: **SPRINGER. Brazilian Symposium on Bioinformatics**. [S.l.], 2012. p. 61–72.
- BRASIL, C. R. S. **Algoritmo evolutivo de muitos objetivos para predição ab initio de estrutura de proteínas**. Tese (Doutorado) — Universidade de São Paulo, 2012.
- BRASIL, C. R. S.; DELBEM, A. C. B.; SILVA, F. L. B. da. Multiobjective evolutionary algorithm with many tables for purely ab initio protein structure prediction. **Journal of computational chemistry**, Wiley Online Library, v. 34, n. 20, p. 1719–1734, 2013.
- CHOU, H. et al. A novel branch-and-bound algorithm for the protein folding problem in the 3d hp model. **IEEE/ACM transactions on computational biology and bioinformatics**, 2019.
- COELLO, C. A. C. C. A short tutorial on evolutionary multiobjective optimization. In: **SPRINGER. International Conference on Evolutionary Multi-Criterion Optimization**. [S.l.], 2001. p. 21–40.
- COHEN, F. E.; KELLY, J. W. Therapeutic approaches to protein-misfolding diseases. **Nature**, Nature Publishing Group, v. 426, n. 6968, p. 905–909, 2003.

- CONSORTIUM, U. Uniprot: the universal protein knowledgebase. **Nucleic acids research**, Oxford University Press, v. 45, n. D1, p. D158–D169, 2016.
- CUTELLO, V.; NARZISI, G.; NICOSIA, G. A multi-objective evolutionary approach to the protein structure prediction problem. **Journal of The Royal Society Interface**, The Royal Society London, v. 3, n. 6, p. 139–151, 2005.
- DEB, K. Multi-objective optimisation using evolutionary algorithms: an introduction. In: **Multi-objective evolutionary optimisation for product design and manufacturing**. [S.l.]: Springer, 2011. p. 3–34.
- DEB, K. et al. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. In: SPRINGER. **International conference on parallel problem solving from nature**. [S.l.], 2000. p. 849–858.
- DILL, K. A. Polymer principles and protein folding. **Protein Science**, Cambridge University Press, v. 8, n. 6, p. 1166–1180, 1999.
- DILL, K. A. et al. Principles of protein folding—a perspective from simple exact models. **Protein science**, Wiley Online Library, v. 4, n. 4, p. 561–602, 1995.
- DILL, K. A. et al. The protein folding problem. **Annu. Rev. Biophys.**, Annual Reviews, v. 37, p. 289–316, 2008.
- DORIGO, M.; BIRATTARI, M. Ant colony optimization. In: **Encyclopedia of machine learning**. [S.l.]: Springer, 2011. p. 36–39.
- DORIGO, M. et al. Ant system: optimization by a colony of cooperating agents. **IEEE Transactions on Systems, man, and cybernetics, Part B: Cybernetics**, v. 26, n. 1, p. 29–41, 1996.
- DUAN, Y.; KOLLMAN, P. A. Pathways to a protein folding intermediate observed in a 1-microsecond simulation in aqueous solution. **Science**, American Association for the Advancement of Science, v. 282, n. 5389, p. 740–744, 1998.
- DUHR, S.; BRAUN, D. Thermophoretic depletion follows boltzmann distribution. **Physical review letters**, APS, v. 96, n. 16, p. 168301, 2006.
- GABRIEL, P. H.; MELO, V. V. d.; DELBEM, A. C. Evolutionary algorithms and hp model for protein structure prediction. **Sba: Controle & Automação Sociedade Brasileira de Automatica**, SciELO Brasil, v. 23, n. 1, p. 25–37, 2012.
- GALVAO, L. C. et al. A new greedy heuristic for 3dhp protein struture prediction with side chain. In: IEEE. **Bioinformatics and Biomedicine Workshops (BIBMW), 2012 IEEE International Conference on**. [S.l.], 2012. p. 77–81.
- GARZA-FABRE, M. et al. Generating, maintaining, and exploiting diversity in a memetic algorithm for protein structure prediction. **Evolutionary computation**, MIT Press, v. 24, n. 4, p. 577–607, 2016.
- GARZA-FABRE, M.; RODRIGUEZ-TELLO, E.; TOSCANO-PULIDO, G. Multiobjectivizing the hp model for protein structure prediction. In: SPRINGER. **European Conference on Evolutionary Computation in Combinatorial Optimization**. [S.l.], 2012. p. 182–193.

- GOLDBERG, D. E.; HOLLAND, J. H. Genetic algorithms and machine learning. **Machine learning**, Springer, v. 3, n. 2, p. 95–99, 1988.
- GREGERSEN, N. et al. Protein misfolding and human disease. **Annu. Rev. Genomics Hum. Genet.**, Annual Reviews, v. 7, p. 103–124, 2006.
- GUO, G. et al. Knn model-based approach in classification. In: SPRINGER. **OTM Confederated International Conferences”On the Move to Meaningful Internet Systems”**. [S.l.], 2003. p. 986–996.
- HADLEY, K. R.; MCCABE, C. Coarse-grained molecular models of water: a review. **Molecular simulation**, Taylor & Francis, v. 38, n. 8-9, p. 671–681, 2012.
- HAJELA, P.; LIN, C.-Y. Genetic search strategies in multicriterion optimal design. **Structural optimization**, Springer, v. 4, n. 2, p. 99–107, 1992.
- HAMMERSLEY, J. **Monte carlo methods**. [S.l.]: Springer Science & Business Media, 2013.
- HART, W. et al. **Protein structure prediction with evolutionary algorithms**. [S.l.], 1999.
- HATTORI, L. T. et al. A novel approach to protein folding prediction based on long short-term memory networks: a preliminary investigation and analysis. In: IEEE. **2018 International Joint Conference on Neural Networks (IJCNN)**. [S.l.], 2018. p. 1–8.
- HATTORI, L. T. et al. A benchmark of optimally folded protein structures using integer programming and the 3d-hp-sc model. **Computational biology and chemistry**, Elsevier, v. 84, p. 107192, 2020.
- HATTORI, L. T. et al. **Inferência de redes de regulação gênica utilizando métodos de busca e otimização**. Dissertação (Mestrado) — Universidade Tecnológica Federal do Paraná, 2016.
- HOLLAND, J. Adaptation in natural and artificial systems: an introductory analysis with application to biology. **Control and artificial intelligence**, University of Michigan Press, 1975.
- HORN, J. rey; NAFPLIOTIS, N.; GOLDBERG, D. E. A niched pareto genetic algorithm for multiobjective optimization. In: CITESEER. **Proceedings of the first IEEE conference on evolutionary computation, IEEE world congress on computational intelligence**. [S.l.], 1994. v. 1, p. 82–87.
- HUANG, C.; YANG, X.; HE, Z. Protein folding simulations of 2d hp model by the genetic algorithm based on optimal secondary structures. **Computational Biology and Chemistry**, Elsevier, v. 34, n. 3, p. 137–142, 2010.
- HUTSON, A. M.; MICKLEBURGH, S. P. **Microchiropteran bats: global status survey and conservation action plan**. [S.l.]: IUCN, 2001.
- I-TASSER. 2019. Disponível em: <<https://zhanglab.ccmb.med.umich.edu/I-TASSER/>>. Acesso em: 8 de novembro de 2009.
- JONES, D. F.; MIRRAZAVI, S. K.; TAMIZ, M. Multi-objective meta-heuristics: An overview of the current state-of-the-art. **European journal of operational research**, Elsevier, v. 137, n. 1, p. 1–9, 2002.

JORGENSEN, W. L.; MAXWELL, D. S.; TIRADO-RIVES, J. Development and testing of the opls all-atom force field on conformational energetics and properties of organic liquids. **Journal of the American Chemical Society**, ACS Publications, v. 118, n. 45, p. 11225–11236, 1996.

JOUNG, I. et al. Conformational space annealing explained: A general optimization algorithm, with diverse applications. **Computer Physics Communications**, Elsevier, v. 223, p. 28–33, 2018.

JR, A. D. M. et al. Charmm: the energy function and its parameterization. **Encyclopedia of computational chemistry**, Wiley Online Library, v. 1, 2002.

KARABOGA, D.; BASTURK, B. On the performance of artificial bee colony (abc) algorithm. **Applied soft computing**, Elsevier, v. 8, n. 1, p. 687–697, 2008.

KENNEDY, R. J. and eberhart, particle swarm optimization. In: **Proceedings of IEEE International Conference on Neural Networks IV**, pages. [S.l.: s.n.], 1995. v. 1000.

KIRKPATRICK, S.; GELATT, C. D.; VECCHI, M. P. Optimization by simulated annealing. **science**, American Association for the Advancement of Science, v. 220, n. 4598, p. 671–680, 1983.

KLEPEIS, J.; FLOUDAS, C. Astro-fold: a combinatorial and global optimization framework for ab initio prediction of three-dimensional structures of proteins from the amino acid sequence. **Biophysical Journal**, Elsevier, v. 85, n. 4, p. 2119–2146, 2003.

KNOWLES, J.; CORNE, D. The pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimisation. In: **Congress on Evolutionary Computation (CEC99)**. [S.l.: s.n.], 1999. v. 1, p. 98–105.

KONAK, A.; COIT, D. W.; SMITH, A. E. Multi-objective optimization using genetic algorithms: A tutorial. **Reliability Engineering & System Safety**, Elsevier, v. 91, n. 9, p. 992–1007, 2006.

KOZA, J. R. **Genetic programming: A paradigm for genetically breeding populations of computer programs to solve problems**. [S.l.]: Stanford University, Department of Computer Science Stanford, CA, 1990.

KURSAWE, F. A variant of evolution strategies for vector optimization. In: **SPRINGER. International Conference on Parallel Problem Solving from Nature**. [S.l.], 1990. p. 193–197.

LEE, J.; FREDDOLINO, P. L.; ZHANG, Y. Ab initio protein structure prediction. In: **From protein structure to function with bioinformatics**. [S.l.]: Springer, 2017. p. 3–35.

LEE, J.; SCHERAGA, H. A.; RACKOVSKY, S. New optimization method for conformational energy calculations on polypeptides: conformational space annealing. **Journal of computational chemistry**, Wiley Online Library, v. 18, n. 9, p. 1222–1232, 1997.

LESK, A. **Introduction to bioinformatics**. [S.l.]: Oxford university press, 2014. 60–304 p.

LI, M. S.; KLIMOV, D.; THIRUMALAI, D. Folding in lattice models with side chains. **Computer physics communications**, Elsevier, v. 147, n. 1-2, p. 625–628, 2002.

LI, X. Protein folding based on simulated annealing algorithm. In: IEEE. **Third International Conference on Natural Computation (ICNC 2007)**. [S.l.], 2007. v. 4, p. 256–259.

LIWO, A. **Computational methods to study the structure and dynamics of biomolecules and biomolecular processes: from bioinformatics to molecular quantum mechanics**. [S.l.]: Springer Science & Business Media, 2013.

LIWO, A. et al. United-residue force field for off-lattice protein-structure simulations: Iii. origin of backbone hydrogen-bonding cooperativity in united-residue potentials. **Journal of Computational Chemistry**, Wiley Online Library, v. 19, n. 3, p. 259–276, 1998.

LIWO, A. et al. A united-residue force field for off-lattice protein-structure simulations. i. functional forms and parameters of long-range side-chain interaction potentials from protein crystal data. **Journal of computational chemistry**, Wiley Online Library, v. 18, n. 7, p. 849–873, 1997.

LIWO, A. et al. A united-residue force field for off-lattice protein-structure simulations. ii. parameterization of short-range interactions and determination of weights of energy terms by z-score optimization. **Journal of computational chemistry**, Wiley Online Library, v. 18, n. 7, p. 874–887, 1997.

LODISH, H. et al. **Biologia celular e molecular**. [S.l.]: Artmed Editora, 2014.

LOPES, H. S. Evolutionary algorithms for the protein folding problem: A review and current trends. In: **Computational intelligence in biomedicine and bioinformatics**. [S.l.]: Springer, 2008. p. 297–315.

MANSOUR, N.; KANJ, F.; KHACHFE, H. Particle swarm optimization approach for protein structure prediction in the 3d hp model. **Interdisciplinary Sciences: Computational Life Sciences**, Springer, v. 4, n. 3, p. 190–200, 2012.

MARUO, M. H.; LOPES, H. S.; DELGADO, M. R. Self-adapting evolutionary parameters: encoding aspects for combinatorial optimization problems. In: SPRINGER. **European Conference on Evolutionary Computation in Combinatorial Optimization**. [S.l.], 2005. p. 154–165.

MATSUMOTO, M.; NISHIMURA, T. Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. **ACM Transactions on Modeling and Computer Simulation (TOMACS)**, ACM New York, NY, USA, v. 8, n. 1, p. 3–30, 1998.

MIRJALILI, S.; MIRJALILI, S. M.; LEWIS, A. Grey wolf optimizer. **Advances in engineering software**, Elsevier, v. 69, p. 46–61, 2014.

MURATA, T.; ISHIBUCHI, H. Moga: Multi-objective genetic algorithms. In: IEEE **international conference on evolutionary computation**. [S.l.: s.n.], 1995. v. 1, p. 289–294.

NELSON, D. L.; COX, M. M. **Princípios de bioquímica de Lehninger**. [S.l.]: Artmed Editora, 2011. 70–229 p.

NUNES, L. F. et al. An integer programming model for protein structure prediction using the 3d-hp side chain model. **Discrete Applied Mathematics**, Elsevier, v. 198, p. 206–214, 2016.

OLIVEIRA, A. J. B. d.; KOIKE, L. An alternative synthesis of 1, 2, 3, 4-tetrahydro-9h-pyrido-[3, 4-b]-indole. **Revista Brasileira de Ciências Farmacêuticas**, SciELO Brasil, v. 39, n. 3, p. 259–264, 2003.

OZBOYACI, M. et al. Modeling and simulation of protein–surface interactions: achievements and challenges. **Quarterly reviews of biophysics**, Cambridge University Press, v. 49, 2016.

PAPOIAN, G. A. **Coarse-grained Modeling of Biomolecules**. [S.l.]: CRC Press, 2017.

PIANA, S.; KLEPEIS, J. L.; SHAW, D. E. Assessing the accuracy of physical models used in protein-folding simulations: quantitative evidence from long molecular dynamics simulations. **Current opinion in structural biology**, Elsevier, v. 24, p. 98–105, 2014.

PIERCE, B. A. **Genética-Um Enfoque Conceitual. Traduzido por Paulo A. Mota, Ex-Professor Adjunto do Departamento de Genética da UFRJ e do Instituto de Biologia da UFF**. [S.l.]: Editora Guanabara Koogan, 2016. 368–373 p.

POIAN, A. D. et al. **Bioquímica I**. Rio de Janeiro: Fundação CECIERJ, 2010. 09–66 p.

QUARK. 2019. Disponível em: <<https://zhanglab.ccmb.med.umich.edu/QUARK/>>. Acesso em: 8 de novembro de 2009.

RCBS. **CSB PROTEIN DATA BANK**. 2020. Disponível em: <<http://www.rcsb.org/pdb/statistics/holdings.do>>. Acesso em: 01 de Novembro de 2020.

ROHL, C. A. et al. Protein structure prediction using rosetta. In: **Methods in enzymology**. [S.l.]: Elsevier, 2004. v. 383, p. 66–93.

ROSETTA. **ROSETTA COMMONS**. 2019. Disponível em: <<https://www.rosettacommons.org/>>. Acesso em: 01 de fevereiro de 2019.

SALOMON-FERRER, R.; CASE, D. A.; WALKER, R. C. An overview of the amber biomolecular simulation package. **Wiley Interdisciplinary Reviews: Computational Molecular Science**, Wiley Online Library, v. 3, n. 2, p. 198–210, 2013.

SCHAFFER, J. D. Multiple objective optimization with vector evaluated genetic algorithms. In: LAWRENCE ERLBAUM ASSOCIATES. INC., PUBLISHERS. **Proceedings of the First International Conference on Genetic Algorithms and Their Applications, 1985**. [S.l.], 1985.

SCHMIDT, T.; BERGNER, A.; SCHWEDE, T. Modelling three-dimensional protein structures for applications in drug design. **Drug discovery today**, Elsevier, v. 19, n. 7, p. 890–897, 2014.

SENIOR, A. W. et al. Improved protein structure prediction using potentials from deep learning. **Nature**, Nature Publishing Group, v. 577, n. 7792, p. 706–710, 2020.

SOTO, C. Protein misfolding and disease; protein refolding and therapy. **FEBS letters**, Elsevier, v. 498, n. 2-3, p. 204–207, 2001.

SOTO, C.; ESTRADA, L. D. Protein misfolding and neurodegeneration. **Archives of neurology**, American Medical Association, v. 65, n. 2, p. 184–189, 2008.

SRINIVAS, N.; DEB, K. Multiobjective optimization using nondominated sorting in genetic algorithms. **Evolutionary computation**, MIT Press, v. 2, n. 3, p. 221–248, 1994.

STORN, R.; PRICE, K. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. **Journal of global optimization**, Springer, v. 11, n. 4, p. 341–359, 1997.

THACHUK, C.; SHMYGELSKA, A.; HOOS, H. H. A replica exchange monte carlo algorithm for protein folding in the hp model. **BMC bioinformatics**, Springer, v. 8, n. 1, p. 342, 2007.

TICONA, W. **Aplicação de algoritmos genéticos multi-objetivo para alinhamento de seqüências biológicas. 2003. 129 f.** Tese (Doutorado) — Dissertação (Mestrado em Ciências)—Instituto de Ciências Matemáticas e de . . . , 2003.

TODORIN, I. et al. A new off-lattice hp model with side-chains for protein folding problem. In: EDP SCIENCES. **ITM Web of Conferences**. [S.l.], 2018. v. 16, p. 02007.

TRAYKOV, M.; ANGELOV, S.; YANEV, N. A new heuristic algorithm for protein folding in the hp model. **Journal of Computational Biology**, Mary Ann Liebert, Inc. 140 Huguenot Street, 3rd Floor New Rochelle, NY 10801 USA, v. 23, n. 8, p. 662–668, 2016.

TRAYKOV, M. et al. Algorithm for protein folding problem in 3d lattice hp model. **International Journal of Biology and Biomedicine**, v. 3, p. 16–21, 2018.

TSUNODA, D. F.; CONSTÂNCIO, A. S. Parameter tuning of a genetic algorithm devised to support automatic protein classification.

UNGER, R.; MOULT, J. Finding the lowest free energy conformation of a protein is an np-hard problem: proof and implications. **Bulletin of mathematical biology**, Springer, v. 55, n. 6, p. 1183–1198, 1993.

VARELA, D.; SANTOS, J. A protein folding model using the face-centered cubic lattice model. In: ACM. **Proceedings of the Genetic and Evolutionary Computation Conference Companion**. [S.l.], 2017. p. 1674–1678.

VENDRUSCOLO, M.; DOBSON, C. M. Protein dynamics: Moore’s law in molecular biology. **Current Biology**, Elsevier, v. 21, n. 2, p. R68–R70, 2011.

VENSKE, S. M.; GONÇALVES, R. A.; DELGADO, M. R. Ademo/d: Multiobjective optimization by an adaptive differential evolution algorithm. **Neurocomputing**, Elsevier, v. 127, p. 65–77, 2014.

VENSKE, S. M. G. S. Predição da estrutura de proteínas off-lattice usando evolução diferencial multiobjetivo adaptativa. Universidade Tecnológica Federal do Paraná, 2014.

WANG, S. et al. Folding membrane proteins by deep transfer learning. **Cell systems**, Elsevier, v. 5, n. 3, p. 202–211, 2017.

WEINER, S. J. et al. A new force field for molecular mechanical simulation of nucleic acids and proteins. **Journal of the American Chemical Society**, ACS Publications, v. 106, n. 3, p. 765–784, 1984.

WILSON, S. R.; CUI, W. Conformation searching using simulated annealing. In: **The Protein Folding Problem and Tertiary Structure Prediction**. [S.l.]: Springer, 1994. p. 43–70.

- YANG, J. et al. The i-tasser suite: protein structure and function prediction. **Nature methods**, Nature Publishing Group, v. 12, n. 1, p. 7, 2015.
- YANG, X.-S. A new metaheuristic bat-inspired algorithm. In: **Nature inspired cooperative strategies for optimization (NICSO 2010)**. [S.l.]: Springer, 2010. p. 65–74.
- YANG, Y. et al. Sixty-five years of the long march in protein secondary structure prediction: the final stretch? **Briefings in bioinformatics**, Oxford University Press, v. 19, n. 3, p. 482–494, 2016.
- YUE, K.; DILL, K. A. Sequence-structure relationships in proteins and copolymers. **Physical Review E**, APS, v. 48, n. 3, p. 2267, 1993.
- YUE, K. et al. A test of lattice protein folding algorithms. **Proceedings of the National Academy of Sciences**, National Acad Sciences, v. 92, n. 1, p. 325–329, 1995.
- ZHANG, B.; CECH, T. R. Peptide bond formation by in vitro selected ribozymes. **Nature**, Nature Publishing Group, v. 390, n. 6655, p. 96, 1997.
- ZHANG, Q.; LI, H. Moea/d: A multiobjective evolutionary algorithm based on decomposition. **IEEE Transactions on evolutionary computation**, IEEE, v. 11, n. 6, p. 712–731, 2007.
- ZHANG, Y. Protein structure prediction: when is it useful? **Current opinion in structural biology**, Elsevier, v. 19, n. 2, p. 145–155, 2009.
- ZHANG, Y.; SKOLNICK, J. Automated structure prediction of weakly homologous proteins on a genomic scale. **Proceedings of the National Academy of Sciences**, National Acad Sciences, v. 101, n. 20, p. 7594–7599, 2004.
- ZHOU, H.; SKOLNICK, J. Ab initio protein structure prediction using chunk-tasser. **Biophysical journal**, Elsevier, v. 93, n. 5, p. 1510–1518, 2007.
- ZITZLER, E.; LAUMANN, M.; THIELE, L. Spea2: Improving the strength pareto evolutionary algorithm. **TIK-report**, Eidgenössische Technische Hochschule Zürich (ETH), Institut für Technische . . . , v. 103, 2001.
- ZITZLER, E.; THIELE, L. An evolutionary algorithm for multiobjective optimization: The strength pareto approach. **TIK-report**, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute . . . , v. 43, 1998.

APÊNDICE A – ESTRUTURAS GERADAS

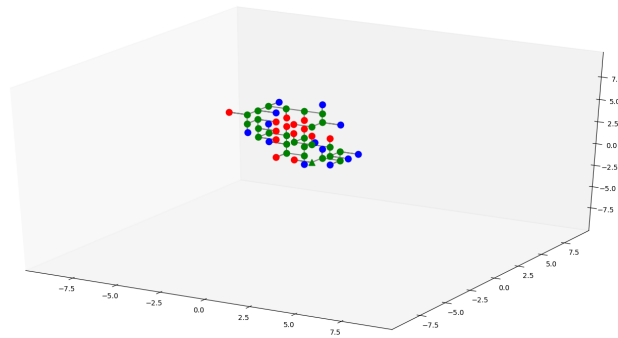
Neste anexo são apresentadas as principais estruturas geradas para todos os experimentos. Nas figuras (a) estão representadas as estruturas tridimensionais das proteínas geradas, sendo:

- O triângulo verde a primeira cadeia lateral, localizada na referência do espaço tridimensional;
- As esferas vermelhas representam as estruturas hidrofóbicas;
- As esferas azuis representam as estruturas hidrofílicas;
- As esferas verdes representam as cadeias laterais e;
- As conexões são as linhas cinzas.

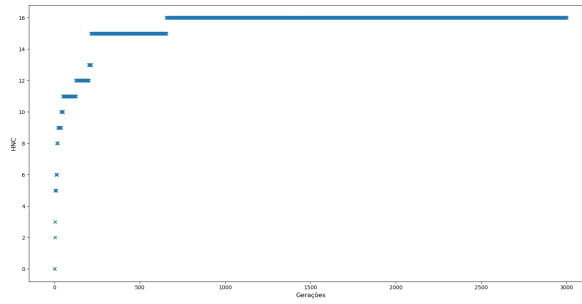
As figuras (b) representam o aumento do número de contatos hidrofóbicos em relação às gerações.

As figuras (c) representam a diminuição da energia livre f_2 em relação às gerações.

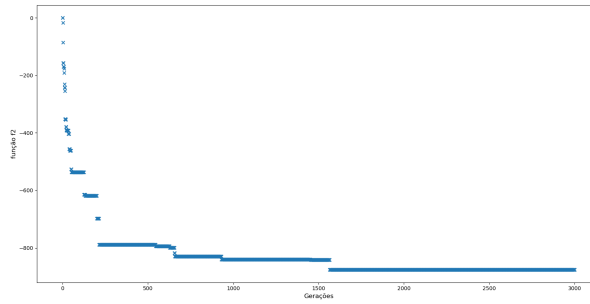
As figuras (d) apresentam a relação entre a diminuição da energia livre f_2 e o número de contatos hidrofóbicos.



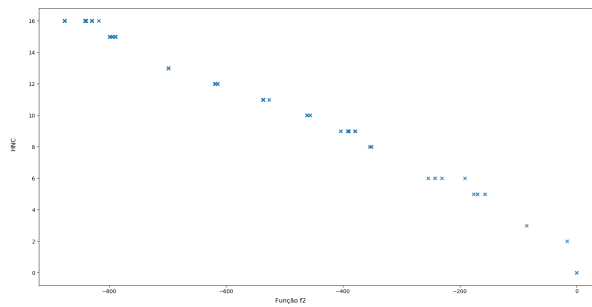
(a) Estrutura Dill.1



(b) HNC X Gerações

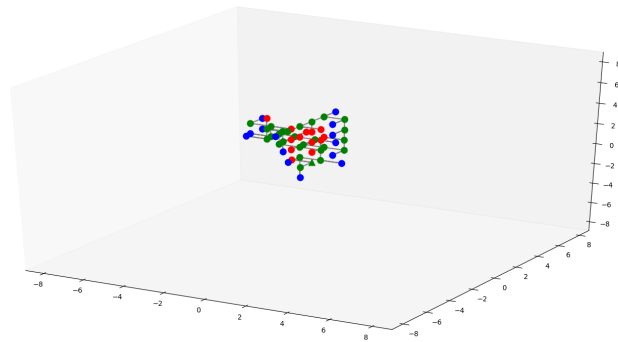


(c) f_2 X Gerações

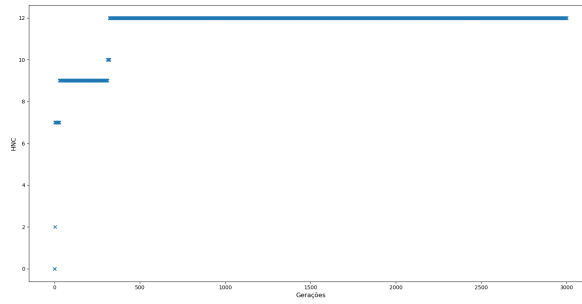


(d) HNC X f_2

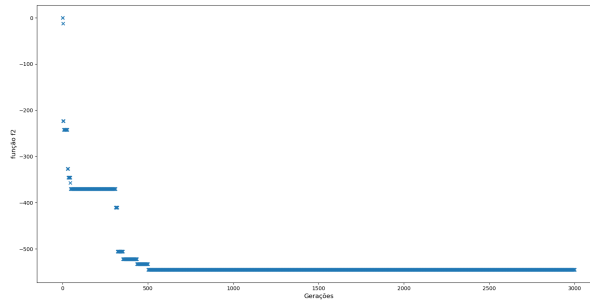
Figura 33: Estrutura e Funções para Dill.1



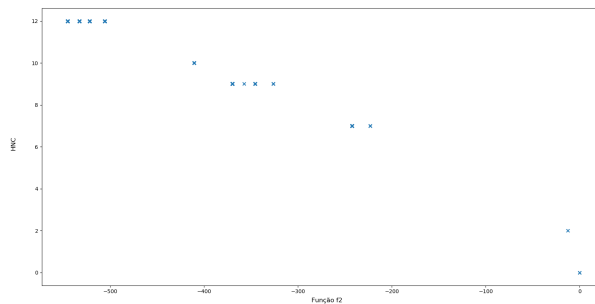
(a) Estrutura Dill.2



(b) HNC X Gerações

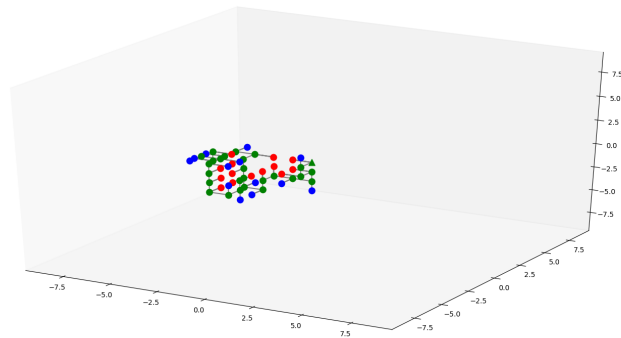


(c) f_2 X Gerações

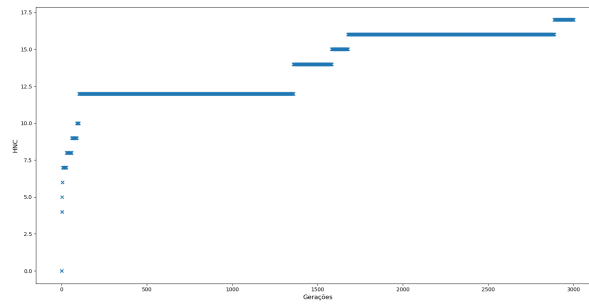


(d) HNC X f_2

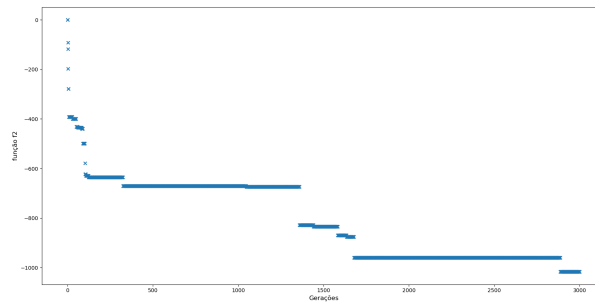
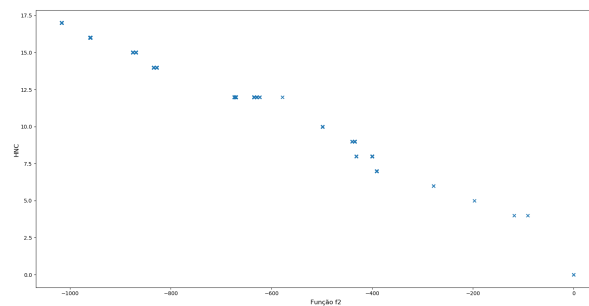
Figura 34: Estrutura e Funções para Dill.2

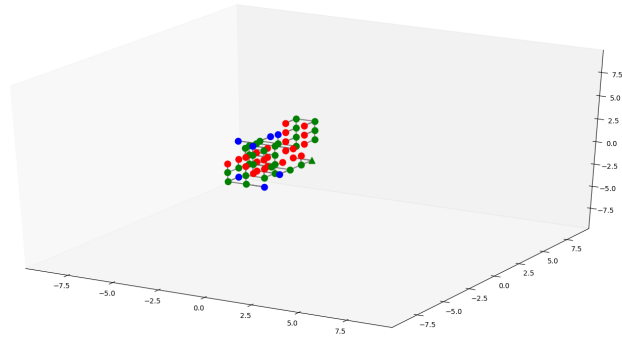


(a) Estrutura Dill.3

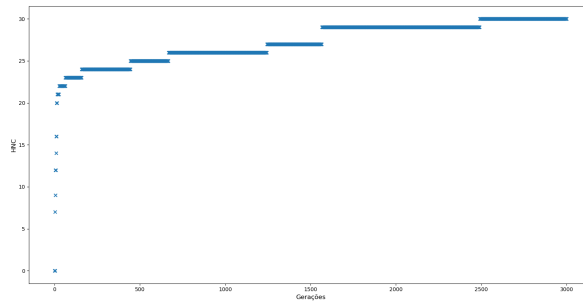


(b) HNC X Gerações

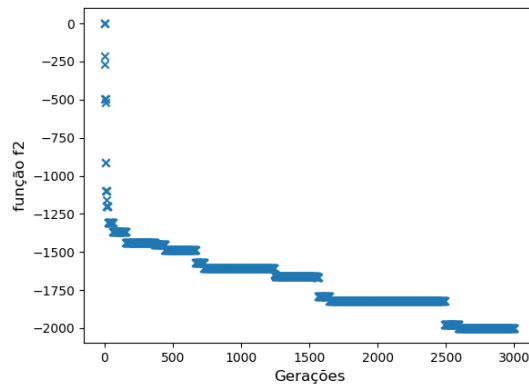
(c) f_2 X Gerações(d) HNC X f_2 **Figura 35: Estrutura e Funções para Dill.3**



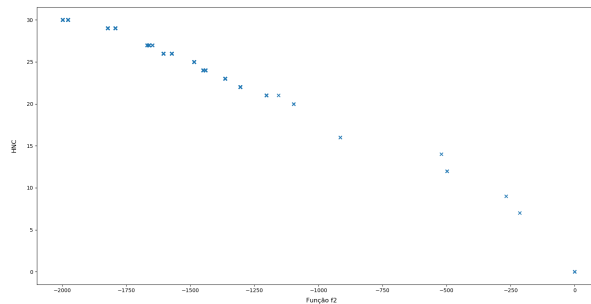
(a) Estrutura Dill.4



(b) HNC X Gerações

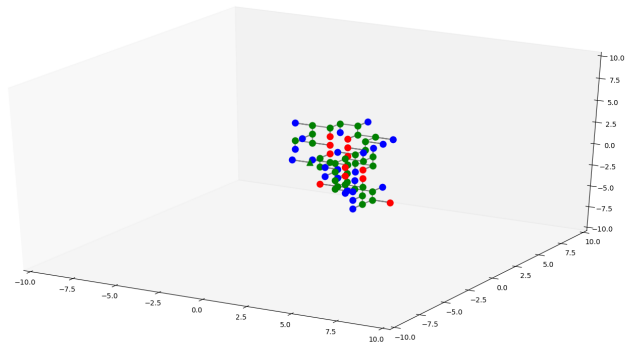


(c) f_2 X Gerações

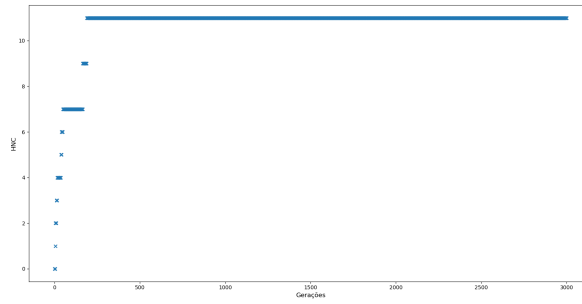


(d) HNC X f_2

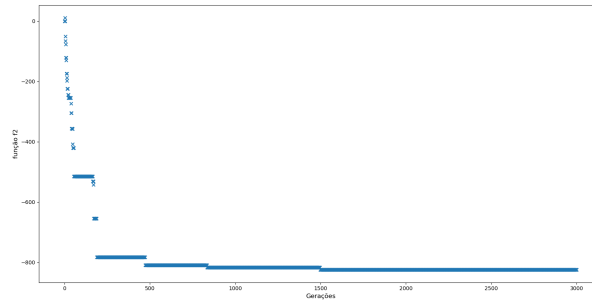
Figura 36: Estrutura e Funções para Dill.4



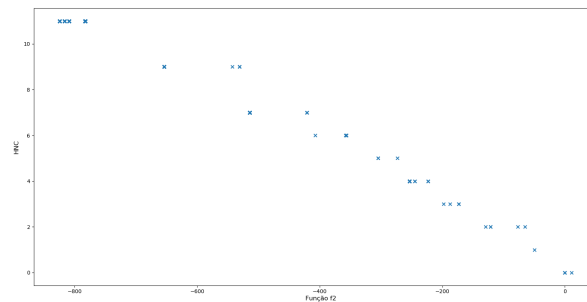
(a) Estrutura Dill.5



(b) HNC X Gerações

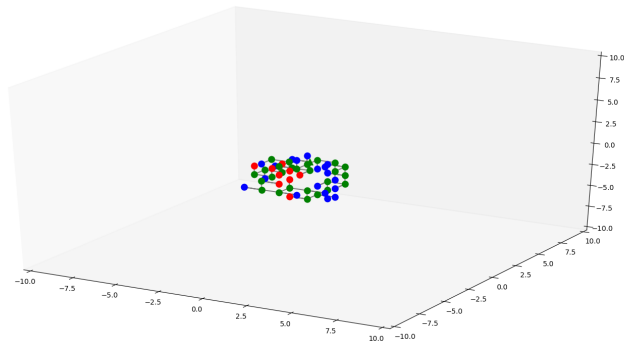


(c) f_2 X Gerações

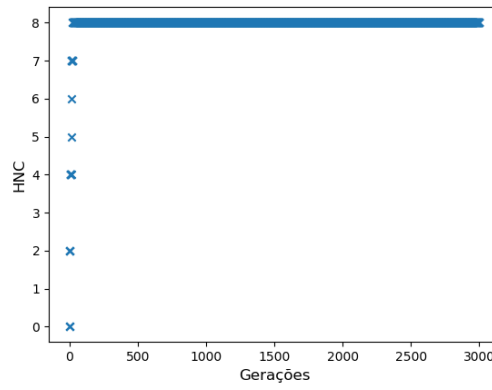


(d) HNC X f_2

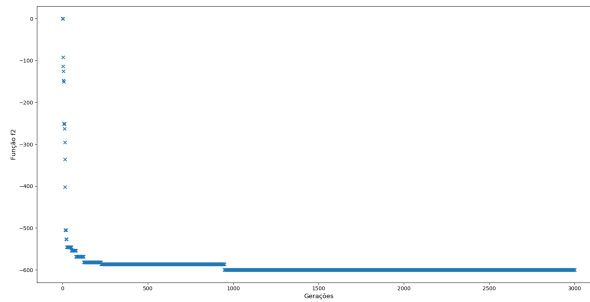
Figura 37: Estrutura e Funções para Dill.5



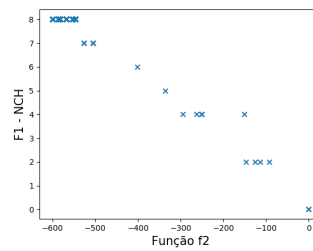
(a) Estrutura Unger273d.1



(b) HNC X Gerações

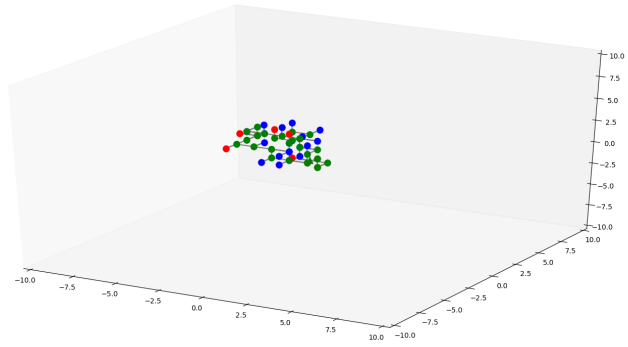


(c) f_2 X Gerações

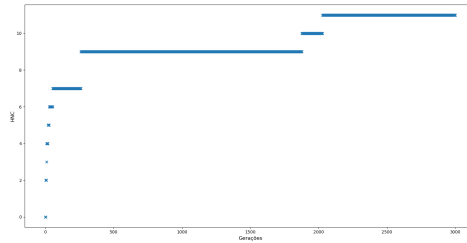


(d) HNC X f_2

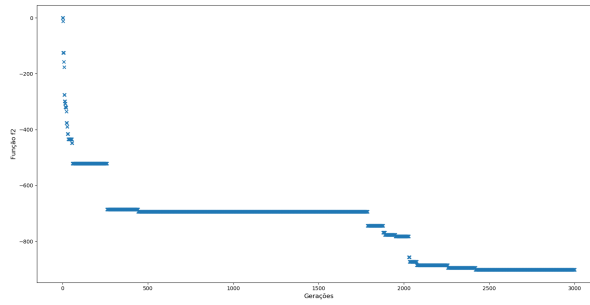
Figura 38: Estrutura e Funções para Unger273d.1



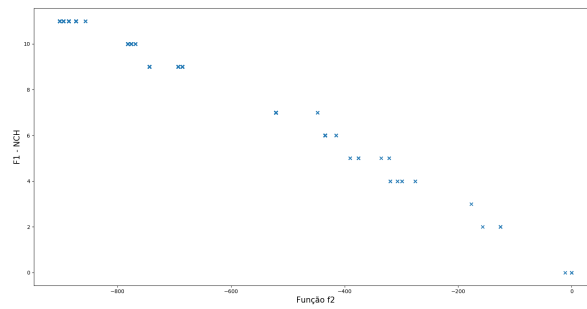
(a) Estrutura Unger273d.3



(b) HNC X Gerações

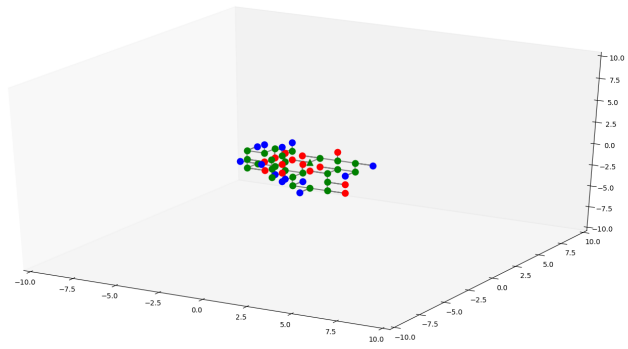


(c) f_2 X Gerações

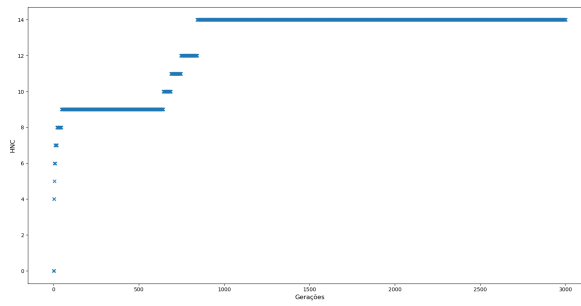


(d) HNC X f_2

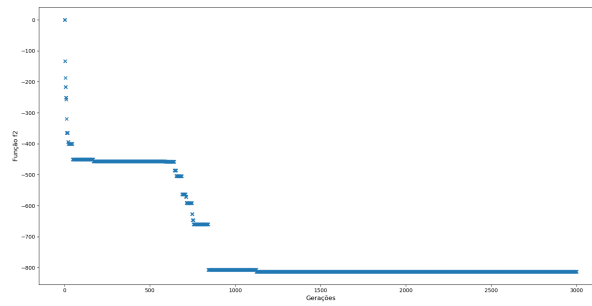
Figura 40: Estrutura e Funções para Unger273d.3



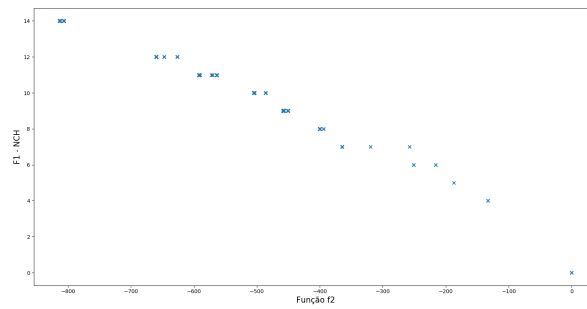
(a) Unger273d.4



(b) HNC X Gerações

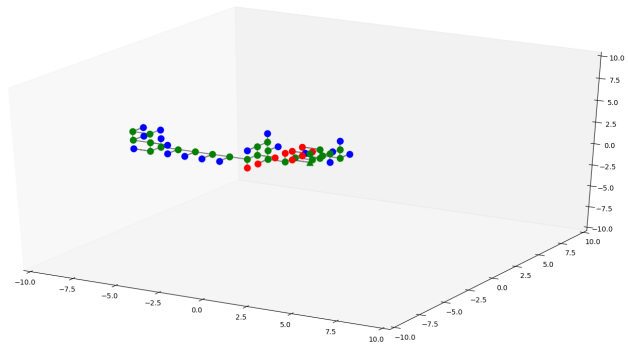


(c) f_2 X Gerações

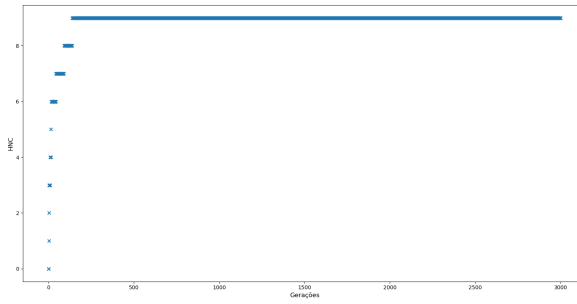


(d) HNC X f_2

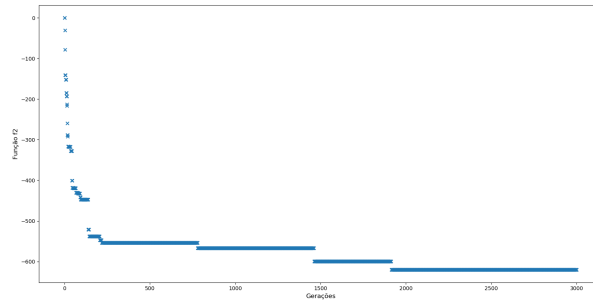
Figura 41: Estrutura e Funções para Unger273d.4



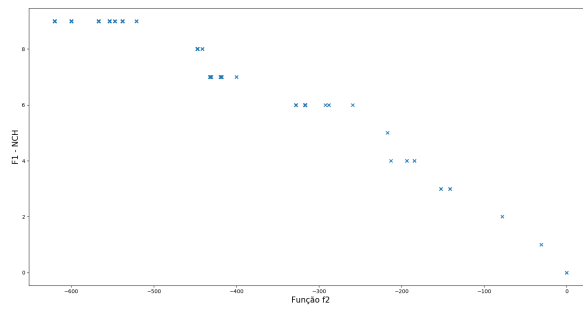
(a) Estrutura Unger273d.5



(b) HNC X Gerações

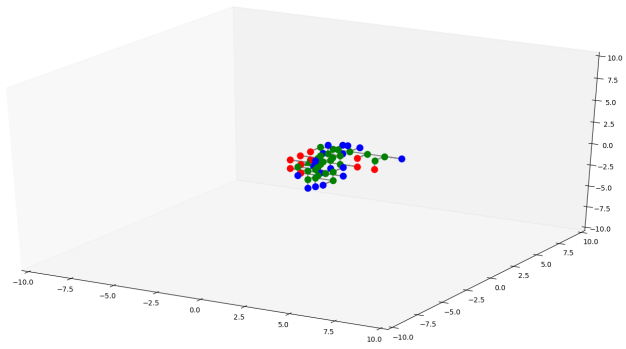


(c) f_2 X Gerações

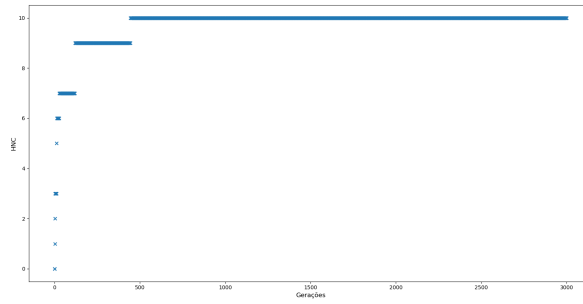


(d) HNC X f_2

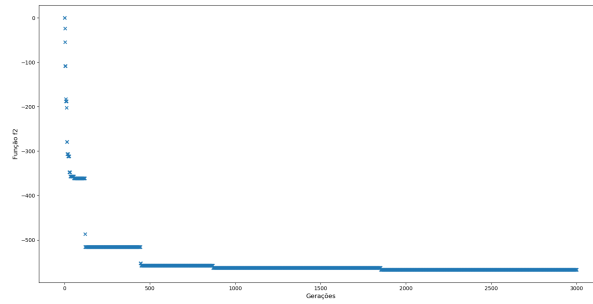
Figura 42: Estrutura e Funções para Unger273d.5



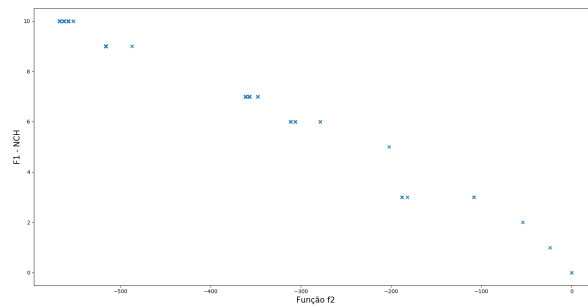
(a) Estrutura Unger273d.6



(b) HNC X Gerações

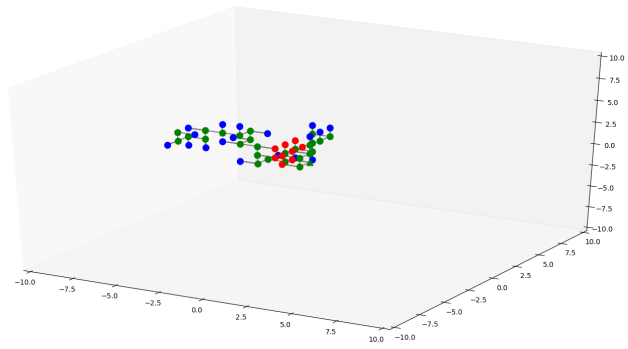


(c) f_2 X Gerações

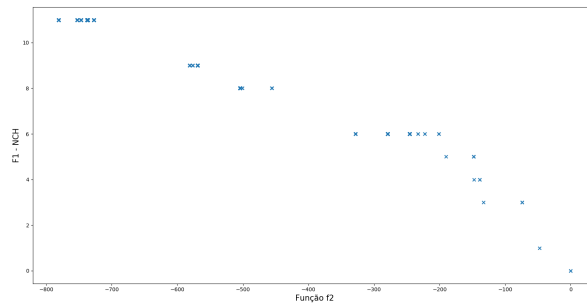


(d) HNC X f_2

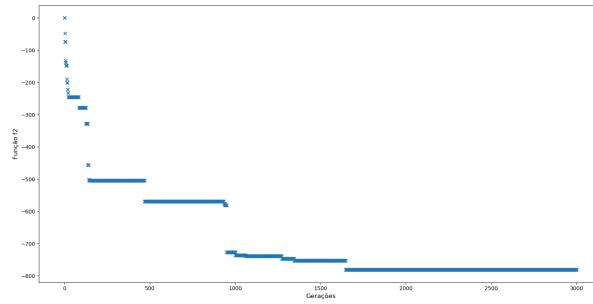
Figura 43: Estrutura e Funções para Unger273d.6



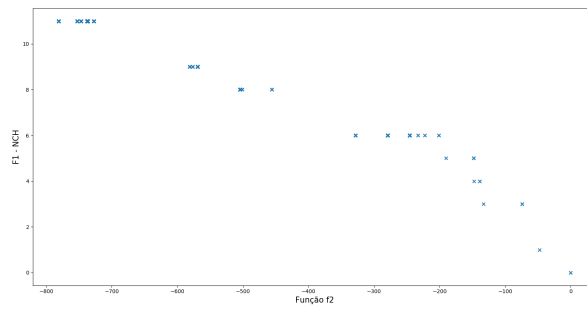
(a) Estrutura Unger273d.7



(b) HNC X Gerações

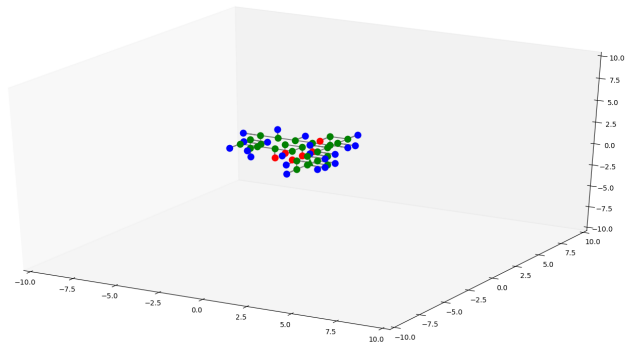


(c) f_2 X Gerações

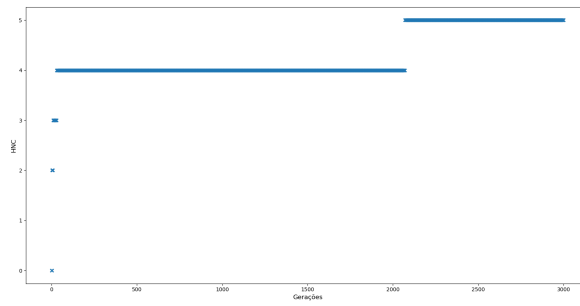


(d) HNC X f_2

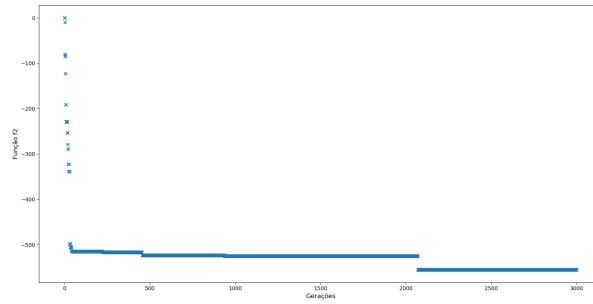
Figura 44: Estrutura e Funções para Unger273d.7



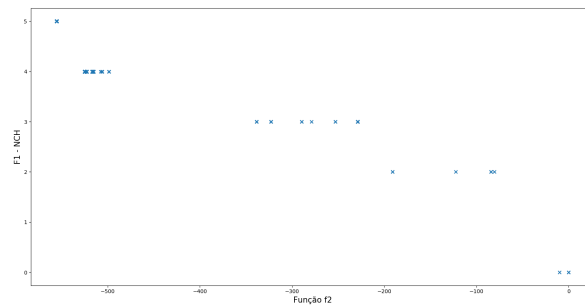
(a) Estrutura Unger273d.8



(b) HNC X Gerações

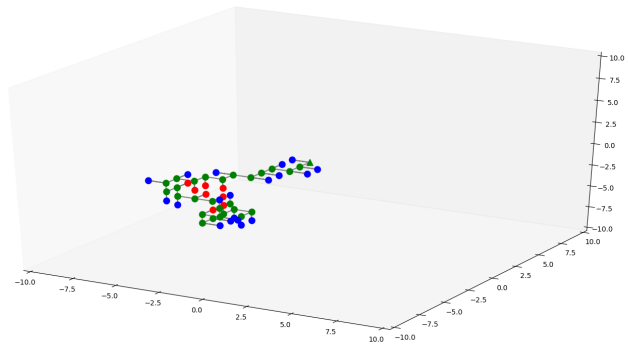


(c) f_2 X Gerações

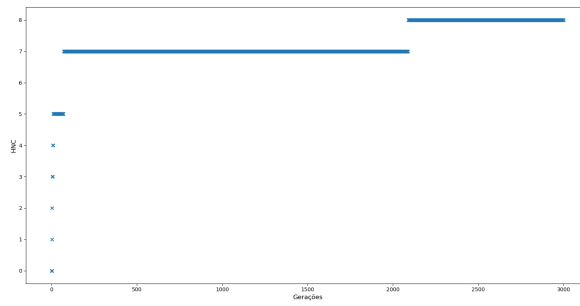


(d) HNC X f_2

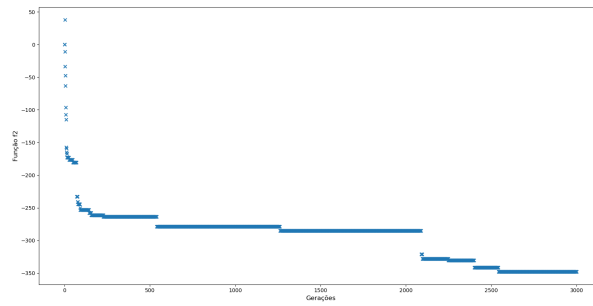
Figura 45: Estrutura e Funções para Unger273d.8



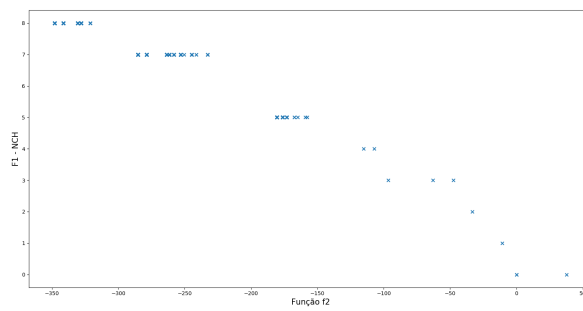
(a) Estrutura Unger273d.8



(b) HNC X Gerações

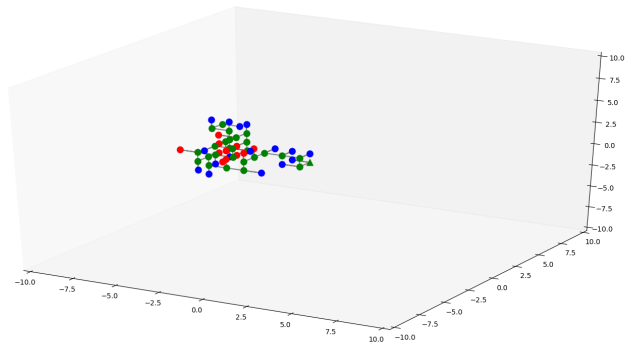


(c) f_2 X Gerações

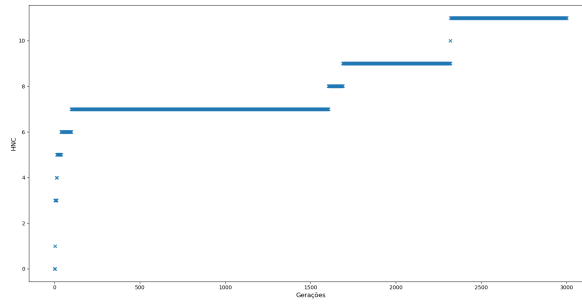


(d) HNC X f_2

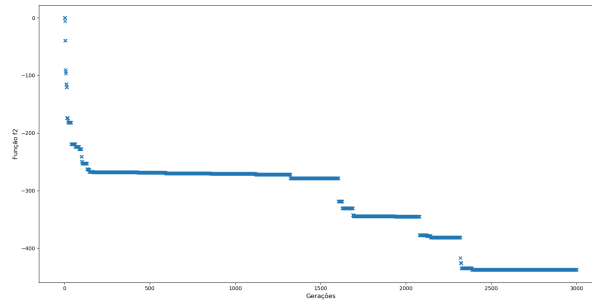
Figura 46: Estrutura e Funções para Unger273d.9



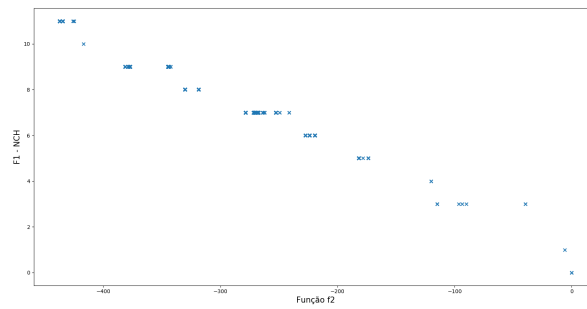
(a) Estrutura Unger273d.10



(b) HNC X Gerações

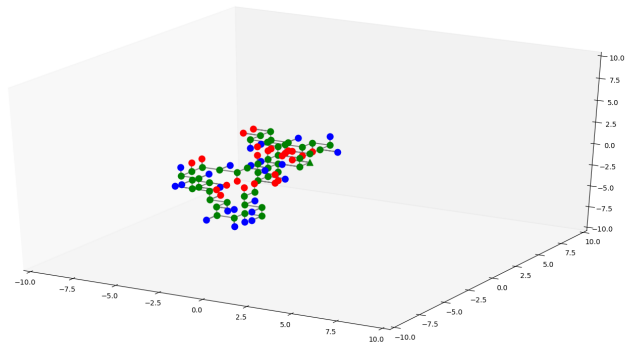


(c) f_2 X Gerações

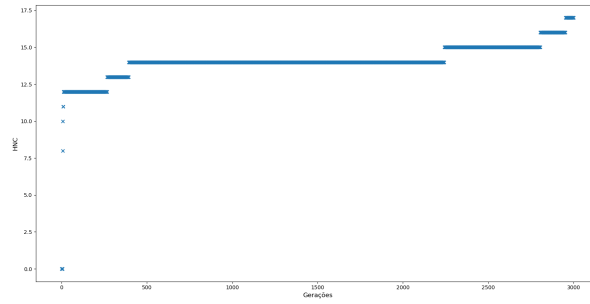


(d) HNC X f_2

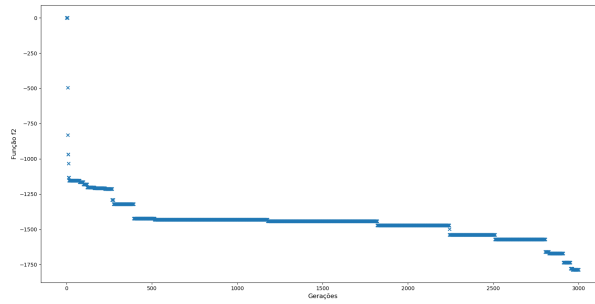
Figura 47: Estrutura e Funções para Unger273d.10



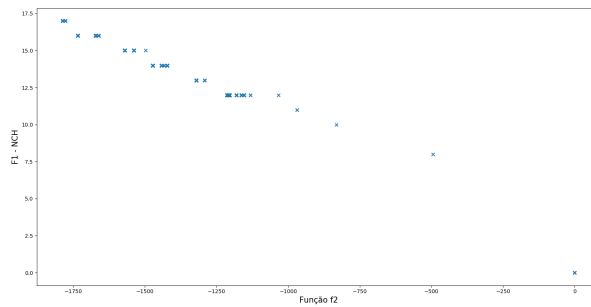
(a) Estrutura S48.1



(b) HNC X Gerações

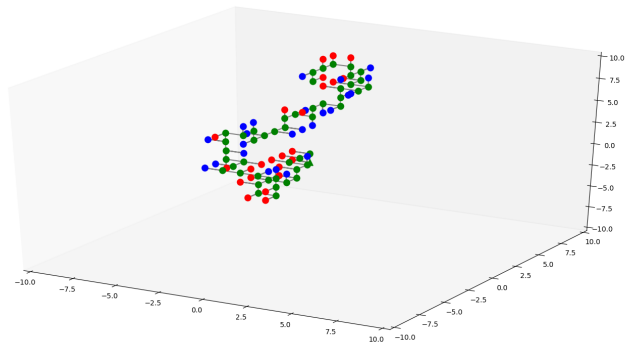


(c) f_2 X Gerações

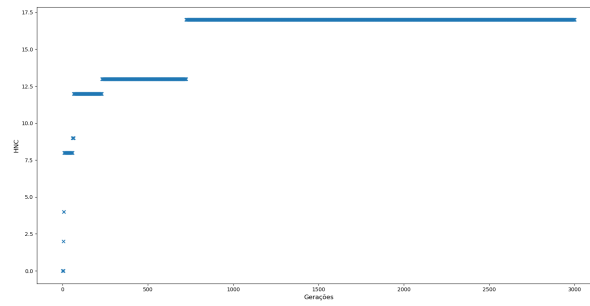


(d) HNC X f_2

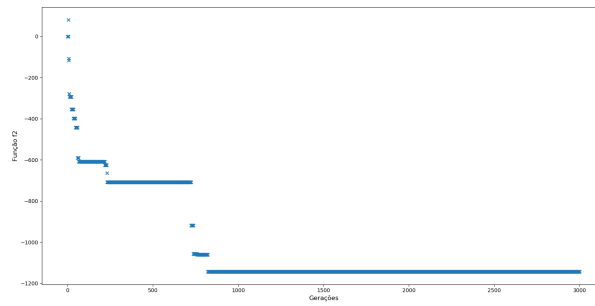
Figura 48: Estrutura e Funções para S48.1



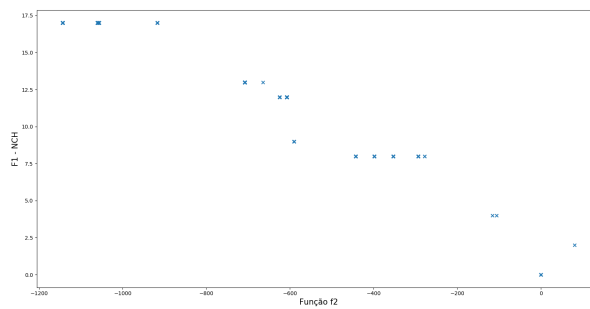
(a) Estrutura S48.2



(b) HNC X Gerações

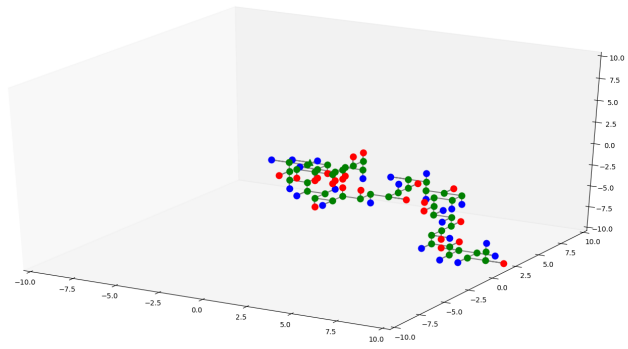


(c) f_2 X Gerações

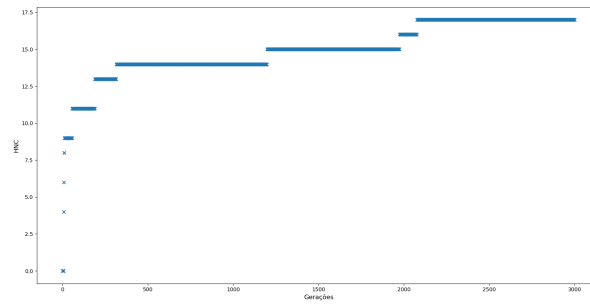


(d) HNC X f_2

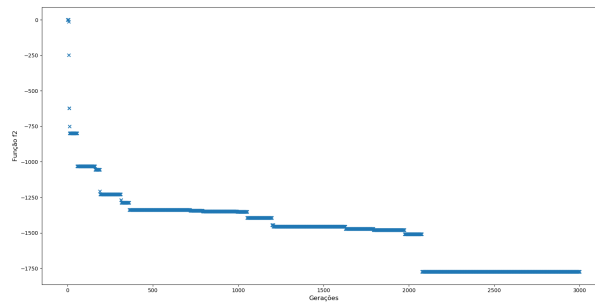
Figura 49: Estrutura e Funções para S48.2



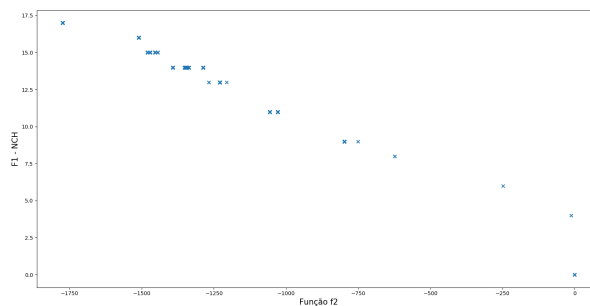
(a) Estrutura S48.3



(b) HNC X Gerações

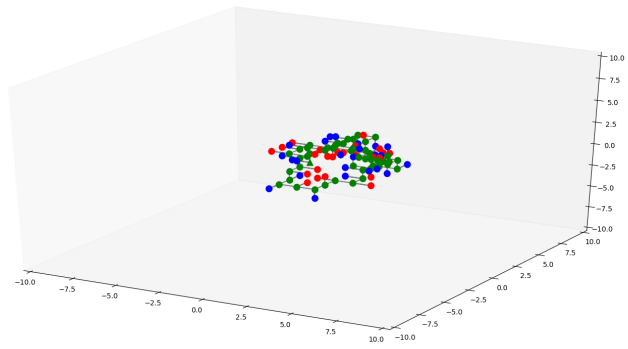


(c) f_2 X Gerações

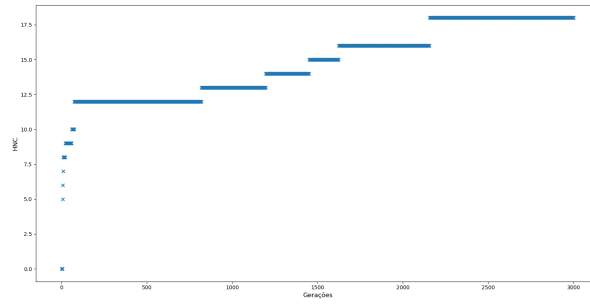


(d) HNC X f_2

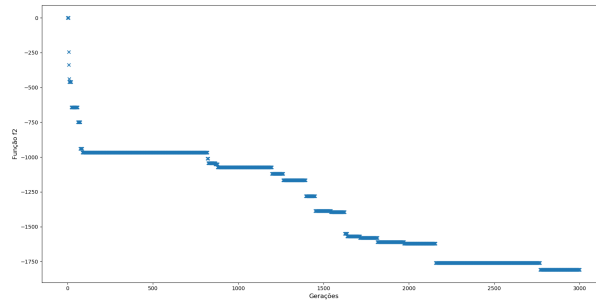
Figura 50: Estrutura e Funções para S48.3



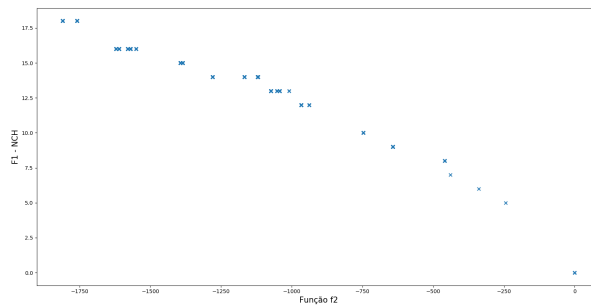
(a) Estrutura S48.4



(b) HNC X Gerações

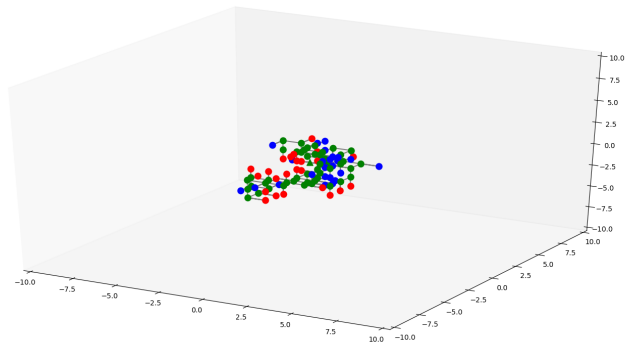


(c) f_2 X Gerações

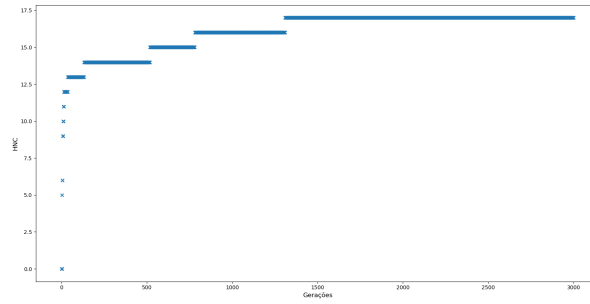


(d) HNC X f_2

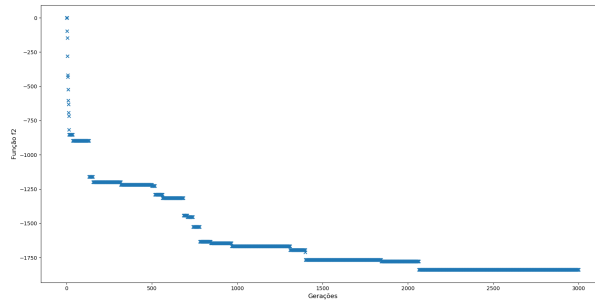
Figura 51: Estrutura e Funções para S48.4



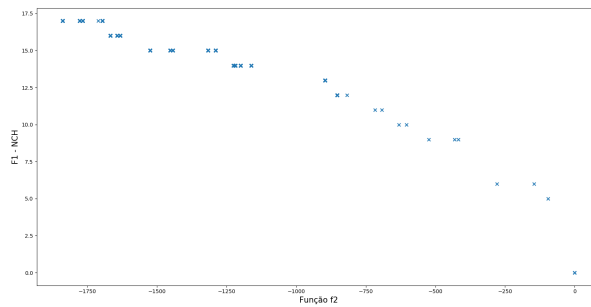
(a) Estrutura S48.5



(b) HNC X Gerações

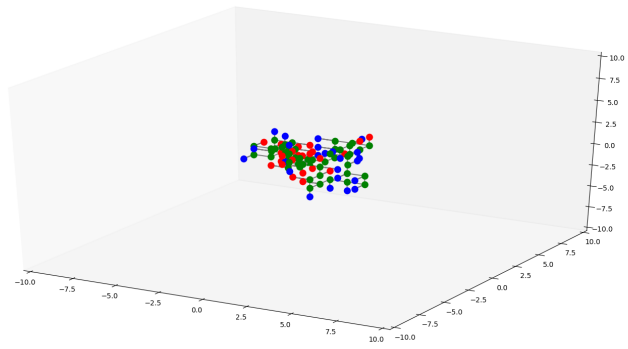


(c) f_2 X Gerações

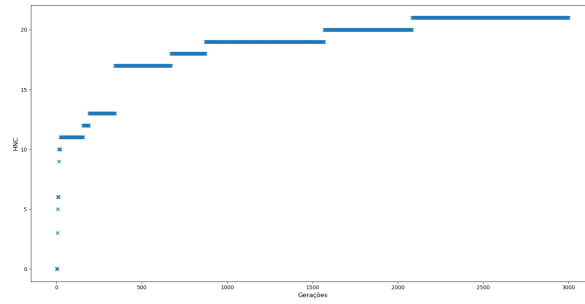


(d) HNC X f_2

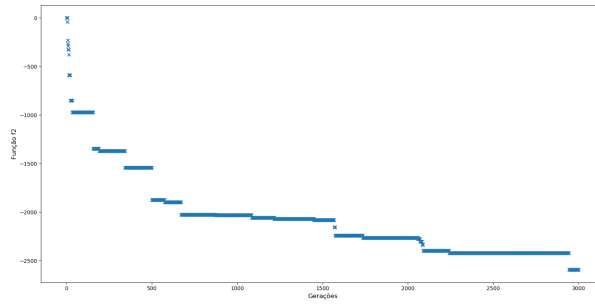
Figura 52: Estrutura e Funções para S48.5



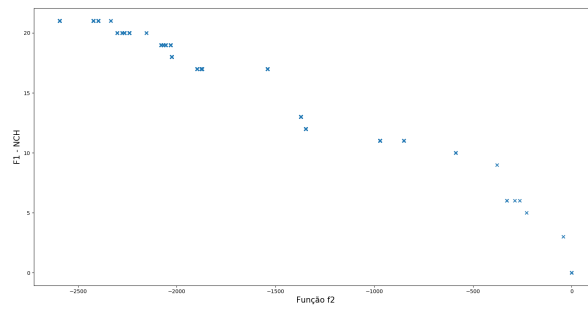
(a) Estrutura S48.6



(b) HNC X Gerações

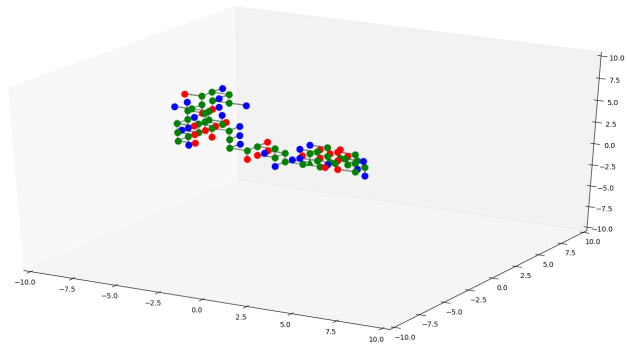


(c) f_2 X Gerações

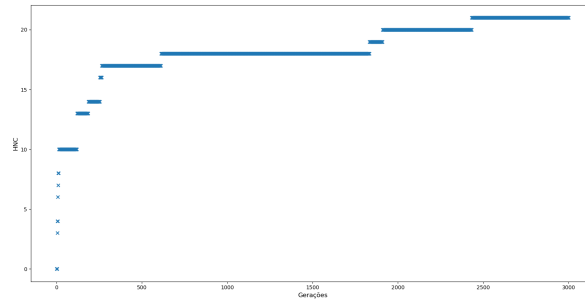


(d) HNC X f_2

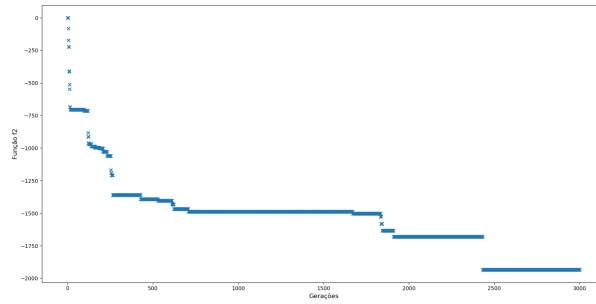
Figura 53: Estrutura e Funções para S48.6



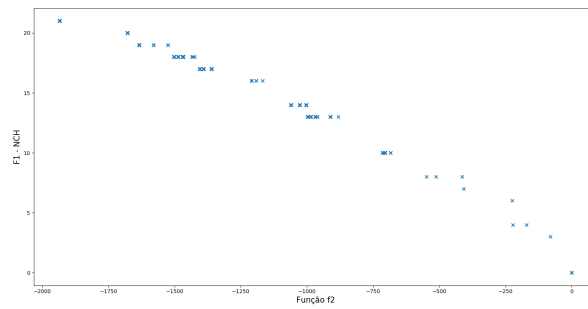
(a) Estrutura S48.7



(b) HNC X Gerações

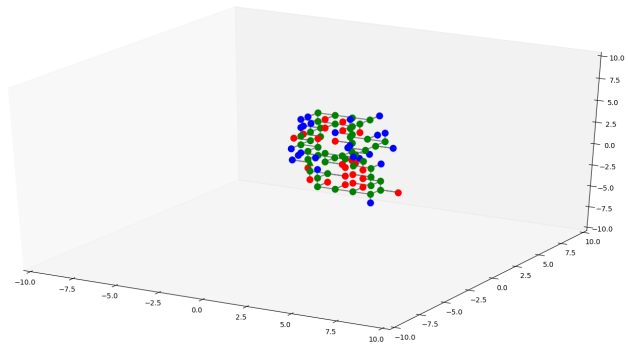


(c) f_2 X Gerações

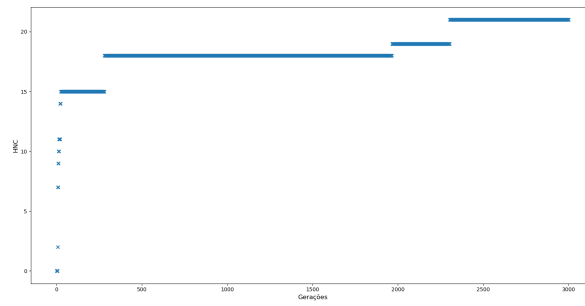


(d) HNC X f_2

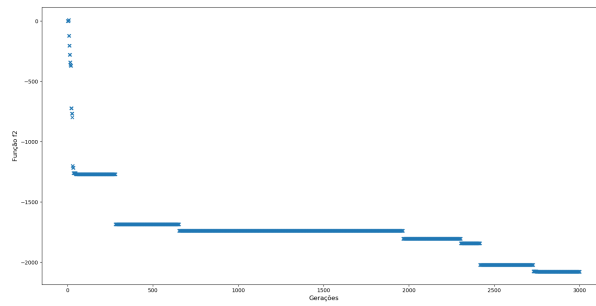
Figura 54: Estrutura e Funções para S48.7



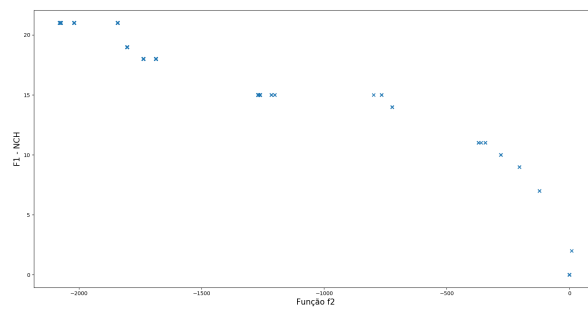
(a) Estrutura S48.8



(b) HNC X Gerações

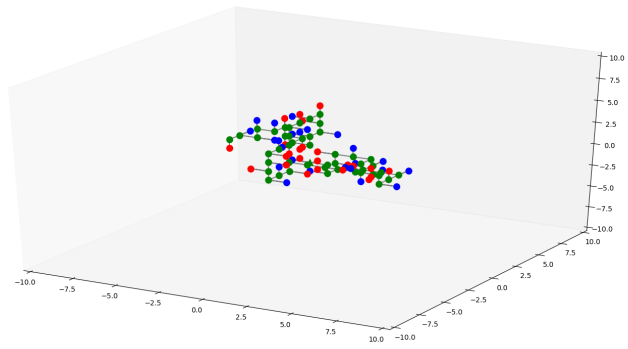


(c) f_2 X Gerações

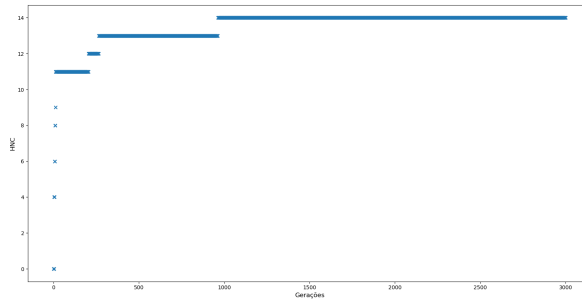


(d) HNC X f_2

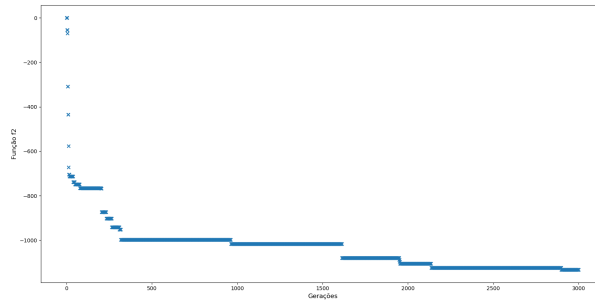
Figura 55: Estrutura e Funções para S48.8



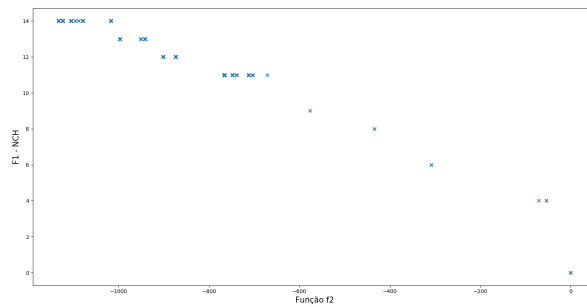
(a) Estrutura S48.9



(b) HNC X Gerações

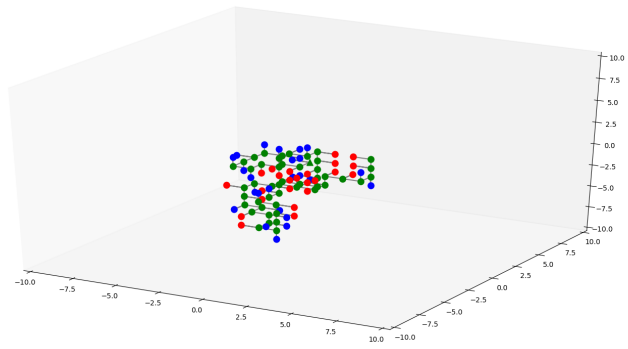


(c) f_2 X Gerações

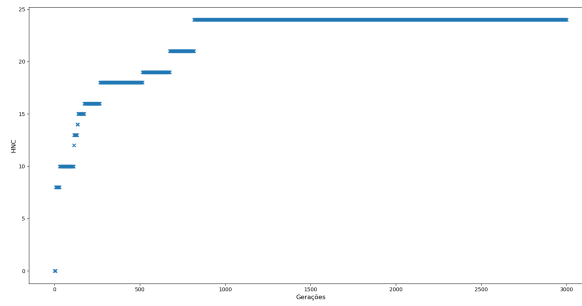


(d) HNC X f_2

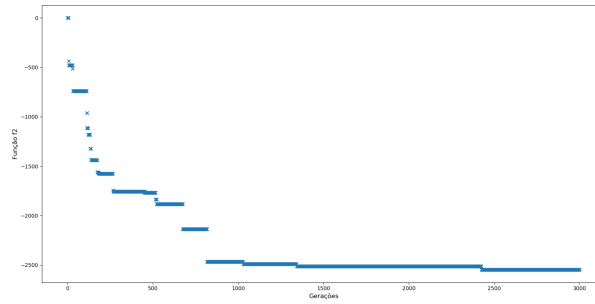
Figura 56: Estrutura e Funções para S48.9



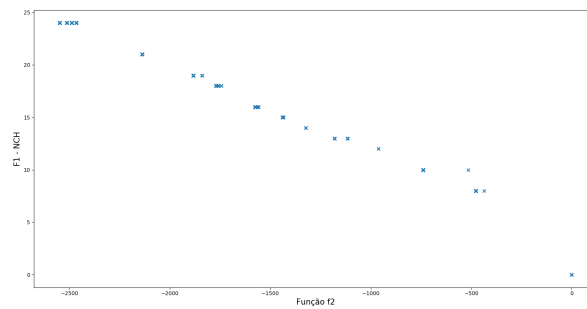
(a) Estrutura S48.10



(b) HNC X Gerações



(c) f_2 X Gerações



(d) HNC X f_2

Figura 57: Estrutura e Funções para S48.10