UNIVERSIDADE FEDERAL DO PARANÁ

LUIZA SCAPINELLO AQUINO DA SILVA

OTIMIZAÇÃO JAYA E VARIANTES APLICADAS AO PROBLEMA DE PREDIÇÃO DE ESTRUTURAS DE PROTEÍNAS

CURITIBA

2021

LUIZA SCAPINELLO AQUINO DA SILVA

OTIMIZAÇÃO JAYA E VARIANTES APLICADAA AO PROBLEMA DE PREDIÇÃO DE ESTRUTURAS DE PROTEÍNAS

Dissertação apresentada ao programa de Pós-Graduação em Engenharia Elétrica, Setor de Tecnologia, Universidade Federal do Paraná, como requisito parcial à obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Prof. Dr. Leandro dos Santos Coelho

CURITIBA

2021

Catalogação na Fonte: Sistema de Bibliotecas, UFPR
Biblioteca de Ciência e Tecnologia

S586o Silva, Luiza Scapinello Aquino da

Otimização jaya e variantes aplicada ao problema de predição de estruturas de proteínas [recurso eletrônico] / Luiza Scapinello Aquino da Silva – Curitiba, 2021.

Dissertação (Mestrado) - Universidade Federal do Paraná, Setor de Tecnologia Programa de Pós-Graduação em Engenharia Elétrica, como requisito parcial à obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Prof. Dr. Leandro dos Santos Coelho

1. Proteínas - Análise. 2. Aminoácidos. I. Coelho, Leandro dos Santos. II. Universidade Federal do Paraná. III. Título.

CDD 518.1

Bibliotecária: Vilma Machado CRB-9/1563

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação ENGENHARIA ELÉTRICA da Universidade Federal do Paraná foram convocados para realizar a arguição da dissertação de Mestrado de **LUIZA SCAPINELLO AQUINO DA SILVA** intitulada: **OTIMIZAÇÃO JAYA E VARIANTES APLICADAS AO PROBLEMA DE PREDIÇÃO DE ESTRUTURAS DE PROTEÍNAS**, sob orientação do Prof. Dr. LEANDRO DOS SANTOS COELHO, que após terem inquirido a aluna e realizada a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

A outorga do título de mestra está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

CURITIBA, 23 de Novembro de 2021.

Assinatura Eletrônica
30/11/2021 09:45:54.0
LEANDRO DOS SANTOS COELHO
Presidente da Banca Examinadora

Assinatura Eletrônica
24/11/2021 10:23:29.0
GIDEON VILLAR LEANDRO
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

Assinatura Eletrônica
24/11/2021 09:41:56.0
MYRIAM REGATTIERI DE BIASE DA SILVA DELGADO
Avaliador Externo (UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ)

Dedico essa pesquisa aos meus pais.

AGRADECIMENTOS

Agradeço primeiramente aos meus pais por sempre terem me apoiado em todas as minhas escolhas e me incentivado nos estudos, agradeço a paciência e por todas as conversas e cafés. Agradeço também ao meu namorado Gabriel por me ajudar a me manter concentrada e acreditar em mim.

Meu agradecimento especial é ao professor Leandro dos Santos Coelho, por todo o apoio e confiança, além de toda a dedicação, tempo e conhecimento que me presenteou. Muito obrigada por ser um ótimo orientador para mim.

Não é no conhecimento que está a felicidade, mas na aquisição do conhecimento. (EDGAR ALLAN POE, 1845)

Uma nova verdade cientifica não triunfa convencendo os opositores e fazendo-os verem a luz, mas sim porque seus opositores eventualmente morrem e uma nova geração cresce que está familiarizada com isso. (MAX PLANK, 1950, p. 33)

RESUMO

O problema de predição de estruturas de proteínas (do inglês *Protein Structure Prediction*, PSP) refere-se ao processo de determinar a sequência de aminoácidos que compõe uma proteína, sendo essa uma área essencial da medicina e biotecnologia. O PSP pode ser abordado como um problema de otimização que visa a determinação da estrutura estável ou nativa de proteínas com mínimo de energia livre possível, sendo o caso de estrutura nativa o foco nessa dissertação. O PSP ainda continua sendo um desafio na bioinformática, devido à falta de acurácia das funções de energia proteicas existentes, além do fato de que o número de sequências de proteínas cresceu exponencialmente, assim como o número de estruturas de proteínas conhecidas. Os bancos de dados atuais, tais como UniProtKB, contêm 93.000.000 de sequências de proteínas computadas, enquanto o Protein Data Bank (PDB) contém aproximadamente 135.000 estruturas conhecidas. Esta desproporcionalidade dos dados torna essa uma área de intensa exploração de abordagens computacionais. Uma das modelagens desse problema, a ab-initio, baseada na hipótese de Anfisen que tenta encontrar a estrutura da proteína a partir da minimização de sua energia livre, tem sido pouco cultivada no desenvolvimento de algoritmos de otimização e aprendizado de máquina, quando comparada a outras modelagens. Leva-se em consideração que mesmo que essa modelagem possua porcentagem de acurácia equivalente às demais, nota-se mais seu uso na literatura apenas para proteínas consideradas de tamanhos reduzidos. Este documento de dissertação de mestrado apresenta uma revisão da literatura em termos de trabalhos relacionados às técnicas utilizadas para resolver este problema, propõe o desenvolvimento de algoritmos que melhor realize o processo de PSP utilizando a modelagem ab-initio em proteínas de mais de 100 aminoácidos de comprimento. Nesse documento de dissertação, a metaheurística de otimização Jaya, inédita nessa aplicação, assim como duas variantes desta, são testadas e avaliadas para o problema de PSP *ab-initio* na modelagem AB *off-lattice*, a qual abstrai a conformação da proteína baseando-se na hidro afinidade de seus aminoácidos. Foram utilizadas sequencias de proteínas tanto reais quando artificiais de diferentes tamanhos retiradas do PDB. Dez sequências de aminoácidos de comprimentos variando de 13 a 143 resíduos foram conformadas pelo algoritmo. Além do mais, experimentos foram realizados com o propósito de avaliar a influência dos hiper parâmetros do algoritmo nos resultados. As conformações finais obtidas mostraram-se como dobramentos bons e coerentes em termos das métricas de análise utilizadas, como o desvio médio quadrático entre os átomos da conformação encontrada e os da proteína original.

Palavras-chave: Predição de estruturas de proteínas. Metaheuristicas. Ab-initio. Otimização Jaya.

ABSTRACT

The Protein Structure Prediction Problem (PSP) refers to the process of determining the sequence of amino acids that make up a protein, which is an essential area of medicine and biotechnology. PSP can be approached as an optimization problem that aims to determine the stable or native structure of proteins with as little free energy as possible, and the case of native structure is the focus of this dissertation. PSP still remains a challenge in bioinformatics due to the lack of accuracy of existing protein energy functions, in addition to the fact that the number of protein sequences has grown exponentially, as has the number of known protein structures. Current databases, such as UniProtKB, contain 93,000,000 computed protein sequences, while the Protein Data Bank (PDB) contains approximately 135,000 known structures. This disproportionality of data makes this an area of intense exploration of computational approaches. One of the models of this problem, *ab-initio*, based on the Anfisen hypothesis that tries to find the protein structure by minimizing its free energy, has been little cultivated in the development of optimization and machine learning algorithms, when compared to other modeling. It is taken into account that even if this modeling has a percentage of accuracy equivalent to the others, its use is more noticeable in the literature only for proteins considered to be of reduced sizes. This master's dissertation document presents a literature review in terms of works related to the techniques used to solve this problem, proposes the development of algorithms that better perform the PSP process using ab-initio modeling in proteins with more than 100 amino acids of length. In this dissertation paper, the Jaya optimization metaheuristic, unprecedented in this application, as well as two variants of it, are tested and evaluated for the PSP *ab-initio* problem in AB off-lattice modeling, which abstracts the protein conformation based on in the hydro affinity of its amino acids. Both real and artificial protein sequences of different sizes taken from the PDB were used. Ten amino acid sequences of lengths ranging from 13 to 143 residues were conformed by the algorithm. Furthermore, experiments were carried out with the purpose of evaluating the influence of the algorithm's hyper parameters on the results. The final conformations obtained proved to be good and coherent folds in terms of the analysis metrics used, such as the root mean square deviation between the atoms of the found conformation and those of the original protein.

Keywords: Prediction of protein structures. Metaheuristics. Ab-initio. Jaya Optimization.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS OU SIGLAS

A	Alanina
aa	Aminoácido
ABC	Otimização de Colônia de Abelhas Artificial (do inglês Artificial Bee Colony)
AEP	Anotações de Estruturas de Proteínas
ANN	Redes Neurais Artificiais (do inglês, *Artificial Neural Network*)
C	Cisteína
C	Carbono
CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
CASP	Avaliação Crítica da Predição de Estruturas de proteínas (do inglês Critical Assessment of protein Structure Prediction)
CNN	Redes Neurais Convolucionais (do inglês, *Convolutional Neural Networks*)
CPU	Unidade Central de Processamento (do inglês, *Central Processing Unit*)
D	Aspartato (ácido aspártico)
DE	Evolução Diferencial (do inglês *Differential Evolution*)
DL	Aprendizado Profundo (do inglês *Deep Learning*)
E	Glutamato (ácido glutâmico)
EA	Algoritmo Evolutivo (do inglês *Evolution Algorithm*)
ES	Estratégias Evolutivas (do inglês *Evolution Strategies*)
EP	Programação Evolutiva (do inglês *Evolutionary Programming*)
F	Fenilalanina
G	Glicina
GA	Algoritmo Genético (do inglês *Genetic Algorithm*)
GP	Programação Genética (do inglês *Genetic Programming*)
H	Histidina
H	Hidrogênio
I	Isoleucina
JayaLF	- Jaya com voo de Lévy (do inglês *Jaya algorithm with Lévy flight*)
K	Lisina
L	Leucina

LJA	Algoritmo Jaya com voo de Lévy (do inglês, *Lévy flight Jaya algorithm*)
M	Metionina
ML	Aprendizado de Máquina (do inglês *Machine Learning*)
MOOP	- Problema de otimização multiobjetivo (do inglês *multi-objective optimization problem*)
N	Asparagina
N	Nitrogênio
NP-difícil	Denota a complexidade polinomial não determinística
O	Oxigênio
P	Prolina
PSP	Predição da Estruturas das Proteínas (do inglês *Protein Structure Prediction*)
PDB	Banco de Dados de Proteínas (do inglês *Protein Data Bank)*
PSO	Otimização por Enxame de Partículas (do inglês *Particle Swarm Optimization)*
Q	Glutamina
R	Arginina
RAM	Memória de Acesso Volátil (do inglês, *Random Access Memory*)
ReLU	Retificador (do inglês *Rectified Linear Unit*)
S	Serina
S	Enxofre
SI	Inteligência de Enxame (do inglês *Swarm Inteligence)*
T	Treonina
UniProtKB	Base de Conhecimento Universal de Proteínas (do inglês *Universal Protein Knowledgebase*)
V	Valina
W	Triptofano
WoS	Web of Science
XGBoost	Gradiente de Impulso Extremo (do inglês *Extreme Gradient Boost)*
Y	Tirosina

SUMÁRIO

1 INTRODUÇÃO

As proteínas são biomoléculas complexas e material fundamental nos organismos vivos, construindo, fortalecendo, mantendo, protegendo e consertando essas entidades. As proteínas são formadas por aminoácidos que são conectados por meio de ligações peptídicas (Jana *et al*., 2018), as proteínas estão envolvidas em todos os processos considerados imprescindíveis para a vida, tal como orientar a catálise de reações bioquímicas, transmissão de sinais e indicar a expressão correta da informação genética (Dhingra *et al*., 2020).

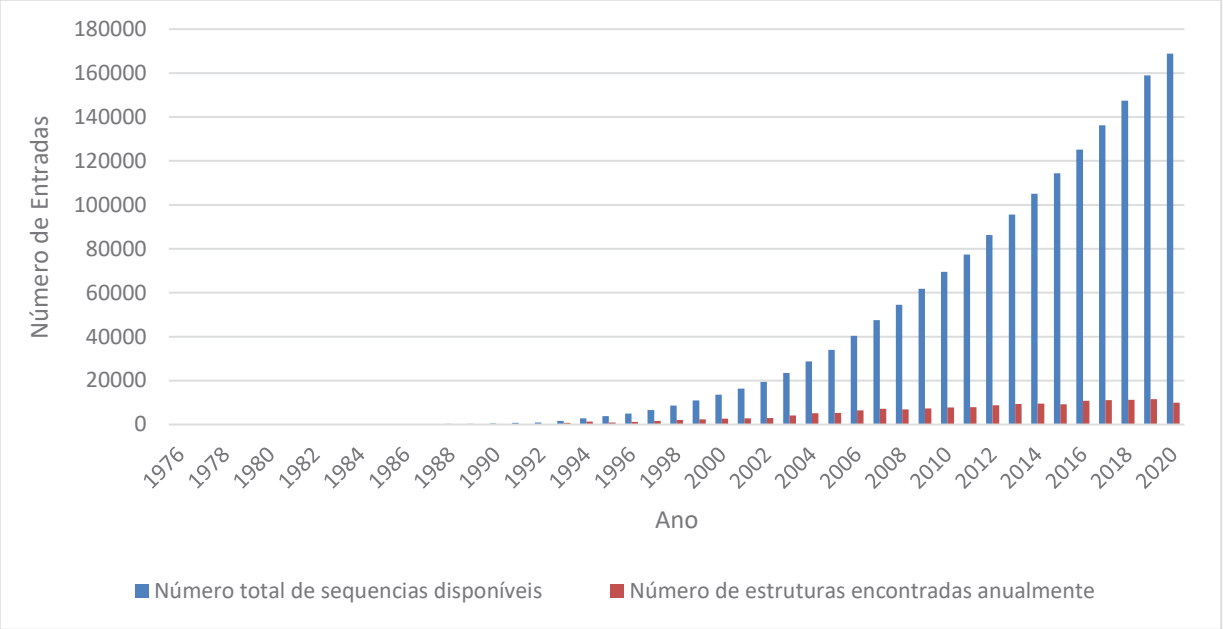
Em sua síntese, a proteína se dobra em uma estrutura tridimensional. Nesse processo, a informação contida na sequência linear de aminoácidos dá origem à conformação tridimensional final bem definida e única. Essa conformação é também chamada de estrutura nativa, a qual acredita-se que seja responsável pela determinação da função da proteína. Dito isso, é possível obter uma previsão da estrutura e função que a proteína desempenhará no organismo utilizando sua sequência linear de aminoácidos.

Contudo, acontece da proteína se moldar incorretamente durante a síntese, gerando uma substância proteica possivelmente prejudicial ao organismo. Algumas doenças correlacionadas a essa má formação são o Alzheimer, Parkinson, fibrose cística, esclerose lateral amiotrófica, câncer e outras doenças menos conhecidas, mas não menos hostis (Jana *et al*., 2018). Isso torna a previsão da estrutura da proteína, por meio da análise das sequências de aminoácidos uma tarefa importante na área da biologia computacional.

Apesar dos avanços tecnológicos recentes destinados à determinação da estrutura de proteínas usando cristalografia de raios-X (Takaya *et al*., 2020), técnicas de espectroscopia por ressonância magnética nuclear (Park *et al*., 2020) e crio-microscopia eletrônica (Nygaard *et al*., 2020), existe ainda uma crescente lacuna entre sequências e estruturas de proteínas conhecidas. Tal avanço levou à necessidade de metodologias de predição de estruturas de proteínas computacionais confiáveis (Kaushik *et al*., 2018).

Nota-se pelo apresentado na Figura 1.1 que o número de sequências conhecidas, encontradas em bancos de dados internacionais tal como o UniProtKB[1], é uma pequena fração da quantidade de estruturas conhecidas do *Protein Data Bank*[2] (PDB, do inglês banco de dados de proteína). Assim como é observado na Figura 1.1, há uma tendência de crescimento exponencial dessa diferença a cada ano.

Figura 1.1 - Comparação das taxas de crescimento anual de sequências de proteínas disponíveis em UniProtKB e estruturas de proteínas disponíveis no PDB.



Fonte: UniProtKB e PDB, 2020.

Devido à grande complexidade do problema de predição de estruturas de proteínas (do inglês *Protein Structure Prediction*, PSP), as metodologias computacionais de metaheurísticas de otimização se destacam bastante, sendo a mais utilizada Algoritmos Genéticos. Contudo, especialmente nos últimos anos, metodologias envolvendo Aprendizado Profundo, como é o caso das Redes Neurais Artificiais e mais especificamente as Redes Neurais Convolucionais se tornaram

---

[1] O UniProt Knowledgebase (UniProtKB) é hub central para a coleta de informações funcionais sobre proteínas, com anotações precisas, consistentes e ricas, de acordo com o seu site oficial.

[2] O arquivo Protein Data Bank (PDB) é o único repositório mundial de informações sobre as estruturas 3D de moléculas biológicas, incluindo proteínas e ácidos nucléicos.

mais populares uma vez que seu desenvolvimento cresceu de forma surpreendente gerando resultados satisfatórios (Das *et al*., 2016).

O objetivo da PSP não é apenas a determinação com precisão da estrutura proteica, mas também determinar os fundamentos biológicos que desencadeiam esse processo (dobramento de proteínas). Consequentemente, será possível compreender melhor o que acontece para que uma proteína saudável se torne irregular e cause doenças.

1.1 JUSTIFICATIVA

Há uma demanda acentuada para decodificar estruturas proteicas como é possível ver pela discrepância de dados sequenciados e estruturas conhecidas por banco de dados como o UniProtKB e o PDB, conforme mencionado no primeiro capítulo.

Consequentemente, existe uma área abrangente de métodos computacionais focados na resolução desse problema, uma vez também que os métodos experimentais laboratoriais são demasiados longos e caros. Metaheurísticas de otimização estão empregados para ajudar a resolver o problema de predição de estruturas de proteínas há mais de duas décadas, contudo, há poucos casos da utilização desses métodos, tanto em conjunto ou separados, quando se trata da modelagem *ab-initio AB Off-Lattice.* Essa modelagem utiliza a equação de energia livre da proteína, que quando minimizada reflete na conformação mais estável, a qual é então traduzida através da modelagem *AB Off-Lattice* para uma representação visual baseada na hidro afinidade dos aminoácidos que compõe tal proteína. Além disso, não há conteúdo na literatura atual do uso de otimização Jaya para PSP.

A modelagem computacional *AB Off-lattice* tem sido amplamente explorada na literatura uma vez que a polaridade é uma das principais forças motrizes que agem no estudo da definição da estrutura da proteína (Jana e Sil, 2020). Portanto, a importância de desenvolver algoritmos de otimização para abordar esse modelo é exposto principalmente em encontrar os núcleos hidrofóbicos de proteínas, consequentemente, identificando proteínas com dobramento incorreto. Do ponto de vista da previsão da estrutura, relacionar a sequência e a estrutura das proteínas é de extrema importância.

1.2 OBJETIVOS

A seguir, o objetivo geral dessa dissertação é abordado, seguido pela descrição dos objetivos específicos.

1.2.1 Objetivo geral

O objetivo geral desta dissertação é desenvolver um algoritmo de abordagem computacional baseado na metaheurística de otimização denominada Jaya aplicado ao problema de PSP *ab-initio AB Off-Lattice*. Comparando assim, esse método, inédito nessa aplicação, com outras abordagens de otimização apresentadas na literatura.

1.2.2 Objetivos específicos

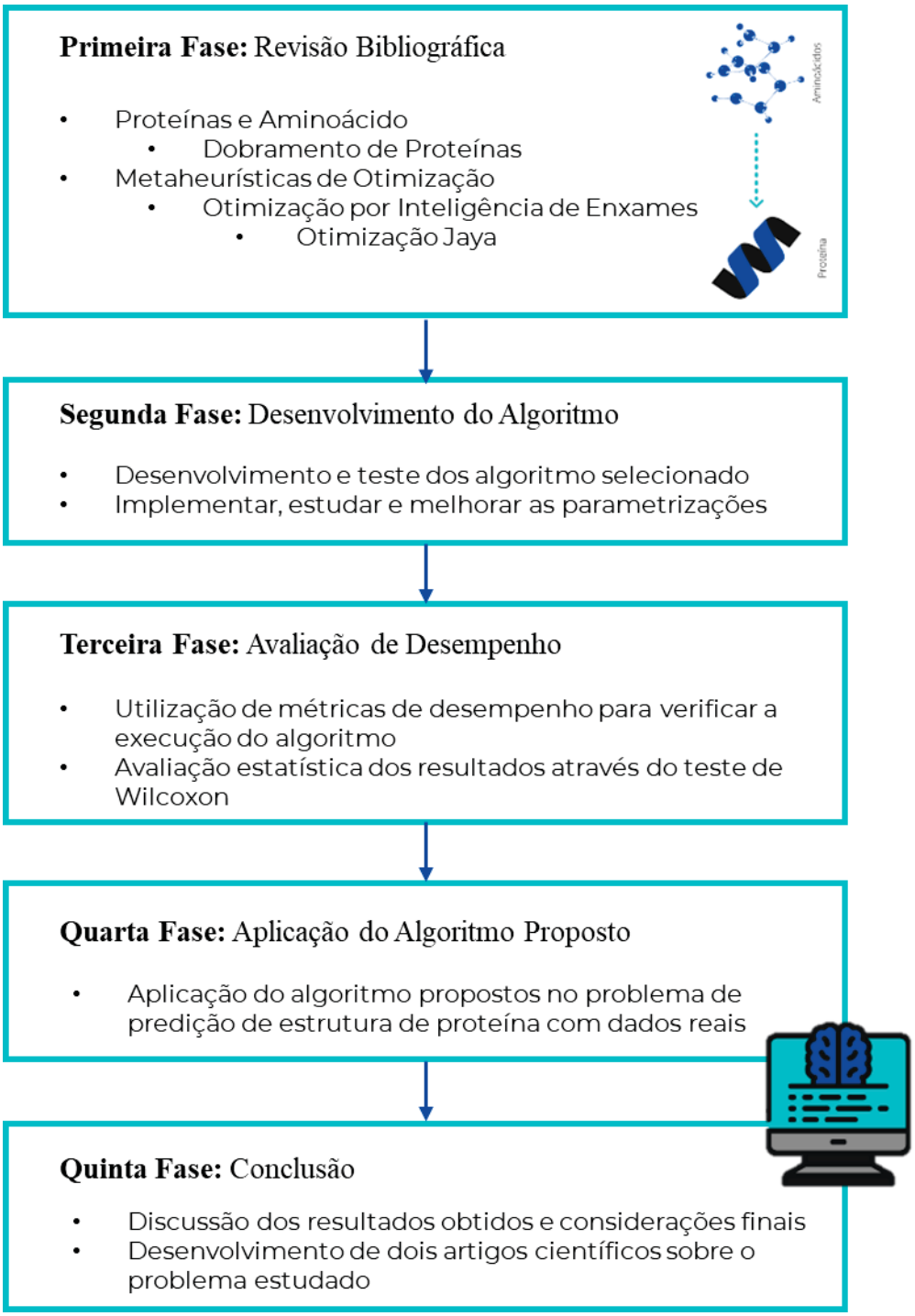
Os objetivos específicos da pesquisa proposta são:

a) Organizar um estudo sobre os fundamentos envolvidos no processo de dobramento das proteínas;
b) Realizar uma revisão da literatura quanto ao uso de métodos computacionais no problema de PSP;
c) Analisar as técnicas mais utilizadas na resolução do problema de predição de estrutura das proteínas com modelagem *ab-initio*;
d) Avaliar a metaheurística de otimização Jaya e duas variantes propostas no problema de PSP;
f) Estudar as influências dos parâmetros do algoritmo nos resultados dos testes, como número de gerações e tamanho da população;
g) Testar os algoritmos tanto com sequências proteicas sintéticas quanto naturais, extraídas do PDB;
h) Avaliar os resultados obtidos utilizando as métricas de desempenho e testes estatísticos.

1.3 METODOLOGIA DE PESQUISA

Para melhor planejar o desenvolvimento da pesquisa, cinco fases foram estipuladas, conforme apresentado na Figura 1.2.

Figura 1.2 - Sequência metodológica da pesquisa



Fonte: a autora, 2020

O primeiro passo envolve conhecer o problema e suas características. O método escolhido para modelar computacionalmente as proteínas é o *ab-initio*, o qual é baseado na função energética de uma proteína. Essa função energética é aplicada então como função objetivo em um problema de otimização para minimizar esse valor. Um dos problemas encontrados no processo de PSP é a obtenção dessa função e para conseguir entendê-la e descobrir qual função melhor se adequa à realidade é necessário fazer um estudo sobre os fundamentos biológicos relacionados à formação e dobramento das proteínas. Embora seja de conhecimento que as proteínas se formam em dobramentos do primário ao secundário, ao terciário e até mesmo aos níveis quaternários, as regras associadas a esse processo permanecem nebulosas (Jana *et al*., 2017).

Posteriormente, a fim de poder adaptar o algoritmo para o problema em questão é necessário entender as características e funcionalidades do programa, é necessário ter e documentar toda a teoria que envolve tal método. Assim é feito uma revisão de algoritmos baseados em metaheurísticas de otimização, mais especificamente também em algoritmos baseados em computação evolutiva e inteligência de enxames, como é o caso da otimização Jaya.

A segunda fase envolve a implementação de testes com o algoritmo a fim de alcançar a parametrização ideal, para que este possa ter o melhor desempenho possível. Na terceira fase, os resultados são relatados pela média, desvio padrão e o melhor implemento em 30 execuções independentes para cada algoritmo. O teste de classificação de Wilcoxon (Carrasco *et al*., 2020) para execuções independentes é conduzido a fim de avaliar se os resultados médios obtidos com o algoritmo de melhor desempenho diferem dos resultados médios do resto dos competidores de uma forma estatisticamente significativa.

Uma outra métrica relevante quando se trata de PSP via métodos computacionais refere-se à comparação da estrutura predita em relação a uma estrutura nativa original, com a finalidade de avaliar quão similar é a adequação predita. Assim, o desvio médio quadrático (do inglês *Root Mean Square Deviation*, RMSD) é frequentemente utilizado, este avalia o grau de semelhança entre as estruturas em *Angstroms* (Å) ou, como é mais comumente utilizado, em nanômetros (nm) (Xu *et al*., 2006).

A quarta fase descreve o algoritmo final executando-o definidamente com os dados dos bancos de dados abertos. Por fim, é apresentada a discussão dos resultados obtidos e as considerações finais.

1.4 ESTRUTURA DO DOCUMENTO

Esta dissertação está organizada em cinco capítulos. O Capítulo 1 introduziu o problema abordado, apresentando os objetivos, justificativa e a metodologia da pesquisa.

No Capítulo 2 são explicados detalhadamente os principais conceitos envolvidos na síntese das proteínas, assim como uma apresentação dos conceitos de metaheurísticas, algoritmos baseados em inteligência de enxames e otimização Jaya, mais especificamente.

No Capítulo 3 é descrito como o algoritmo foi desenvolvido, os detalhes dos bancos de dados utilizados e os principais conceitos sobre avaliação de desempenho, inferência estatística e análise de aptidão.

No Capítulo 4 os resultados obtidos e suas comparações são abordados. Finalmente, o Capítulo 5 apresenta a discussão dos resultados, as conclusões da dissertação e as propostas de pesquisas futuras.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta uma explicação abrangente do problema de previsão de estruturas de proteínas. Primeiramente, explorando os conceitos básicos e biológicos envolvidos na composição de uma proteína, o processo de dobramento proteico é descrito formalmente. Os modelos computacionais, assim como as técnicas utilizadas para resolver esse problema também são discutidos, fundamentando-se nos conceitos básicos desses algoritmos incluindo metaheurísticas de otimização e modelos de aprendizado de máquina.

2.1 PROTEÍNAS E AMINOÁCIDOS

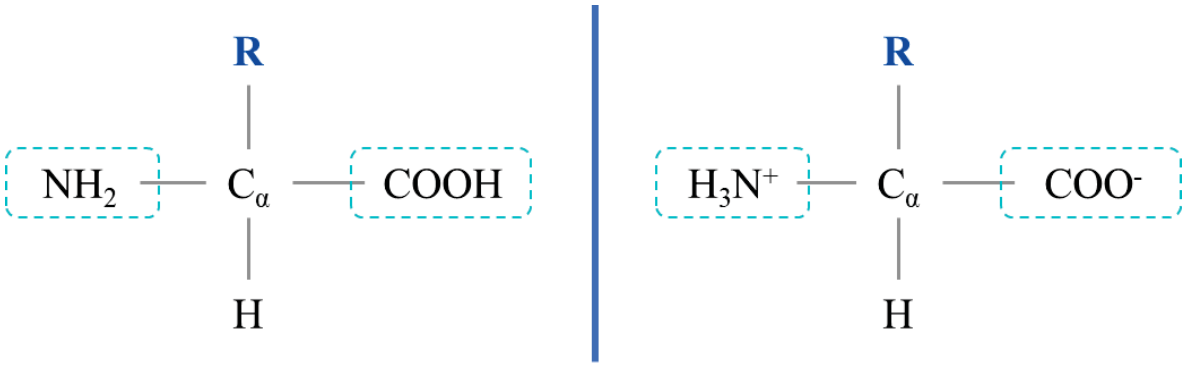
As proteínas são encontradas em todas as células, assim como em cada parte de uma célula, as proteínas são estimadas como as macromoléculas complexas e orgânicas mais abundantes encontradas em seres vivos. Muitos tipos diferentes podem ser observados em uma única célula, consequentemente, existe uma quantidade quase infinita de funções que uma proteína pode oferecer, estando presente em praticamente todos os processos ocorridos dentro de uma célula (NELSON e COX, 2014).

A unidade estrutural base que forma as proteínas é o aminoácido, o qual é descrito como um composto químico formado a partir de um grupo amina (NH_2), um grupo carboxila (COOH) e um grupo único R (Jana *et al.*, 2018). A Figura 2.1 representa a estrutura geral de um aminoácido, comum a todos os chamados de α-aminoácidos[3], especificando aqui que o grupo R, também conhecido como cadeia lateral, ligado ao carbono α é diferente para cada aminoácido.

---

[3] Existe uma exceção a esta regra, que é a prolina, um aminoácido cíclico.

Figura 2.1 – Duas representações da estrutura geral de um aminoácido



Fonte: a autora, 2020.

Todas as proteínas são formadas a partir de um conjunto onipresente de vinte aminoácidos. Eles se diferem em tamanho, forma, capacidade de ligação com um hidrogênio de carga, hidrofobia e reatividade, características assinaladas por seu grupo R exclusivo (Ferina e Daggett, 2019). Assim, as células produzem proteínas com qualidades e funções distintas ligando os mesmos vinte aminoácidos em combinações e sequências diferentes, criando componentes orgânicas como enzimas, hormônios, anticorpos e até fibras musculares. A Tabela 2.1 mostra quais são os vinte aminoácidos presentes na composição das proteínas.

Tabela 2.1 Aminoácidos Essenciais

Nome	Abreviação	Fórmula Química	Hidro afinidade
Alanina	A	CH_3-CH(NH_2)-COOH	Hidrofóbico
Arginina	R	HN=C(NH_2)-NH-(CH_2)$_3$-CH(NH_2)-COOH	Hidrofílico
Asparagina	N	H_2N-CO-CH_2-CH(NH_2)-COOH	Hidrofílico
Aspartato (Ácido Aspártico)	D	HOOC-CH_2-CH(NH_2)-COOH	Hidrofílico
Cisteína	C	HS-CH_2-CH(NH_2)-COOH	Hidrofóbico
Glutamina	Q	H_2N-CO-(CH_2)$_2$-CH(NH_2)-COOH	Hidrofílico
Glutamato (Ácido Glutâmico)	E	HOOC-(CH_2)$_2$-CH(NH_2)-COOH	Hidrofílico
Glicina	G	NH_2-CH_2-COOH	Hidrofóbico
Histidina	H	NH-CH=N-CH=C-CH_2-CH(NH_2)-COOH	Hidrofílico
Isoleucina	I	CH_3-CH_2-CH(CH_3)-CH(NH_2)-COOH	Hidrofóbico
Leucina	L	(CH_3)$_2$-CH-CH_2-CH(NH_2)-COOH	Hidrofóbico
Lisina	K	H_2N-(CH_2)$_4$-CH(NH_2)-COOH	Hidrofílico
Metionina	M	H_2N-(CH_2)$_4$-CH(NH_2)-COOH	Hidrofóbico
Fenilalanina	F	Ph-CH_2-CH(NH_2)-COOH	Hidrofílico
Prolina	P	NH-(CH_2)$_3$-CH-COOH	Hidrofóbico

Serina	S	$HO\text{-}CH_2\text{-}CH(NH_2)\text{-}COOH$	Hidrofílico
Treonina	T	$CH_3\text{-}CH(OH)\text{-}CH(NH_2)\text{-}COOH$	Hidrofílico
Triptofano	W	$Ph\text{-}NH\text{-}CH{=}C\text{-}CH_2\text{-}CH(NH_2)\text{-}COOH$	Hidrofílico
Tirosina	Y	$HO\text{-}p\text{-}Ph\text{-}CH_2\text{-}CH(NH_2)\text{-}COOH$	Hidrofílico
Valina	V	$(CH_3)_2\text{-}CH\text{-}CH(NH_2)\text{-}COOH$	Hidrofóbico

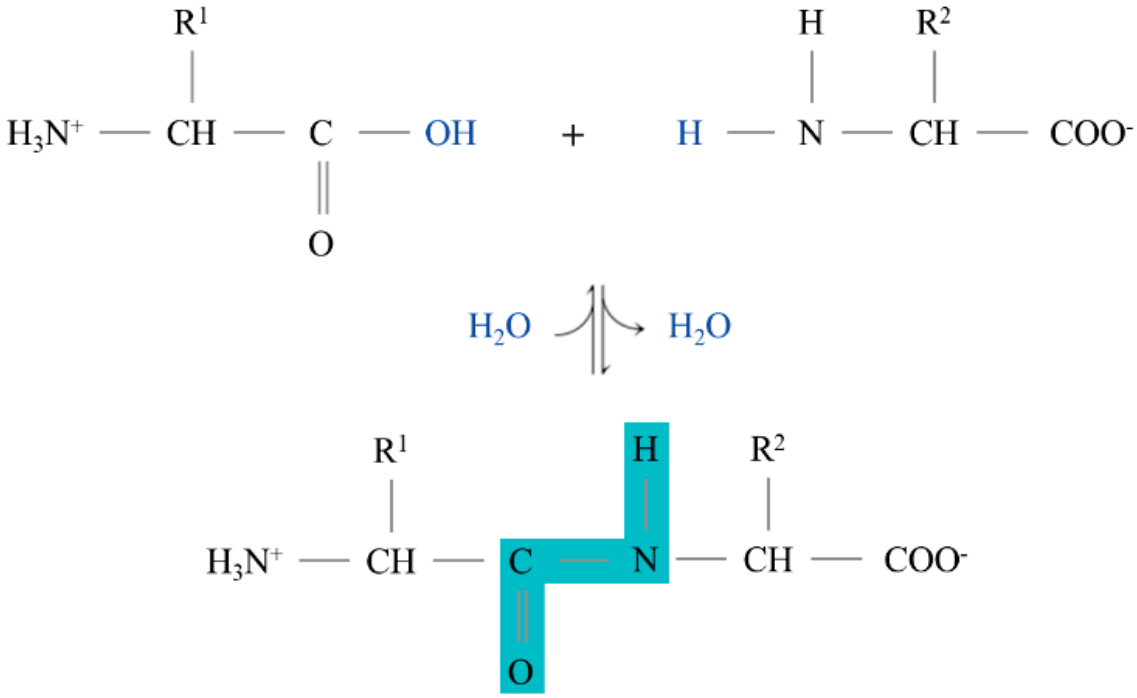
Fonte: a autora, 2021

Os aminoácidos se ligam covalentemente em uma sequência linear característica para criar uma proteína, chamada de ligação peptídica. Nela, o grupo carboxila de um aminoácido se junta ao grupo amina de outro e a reação produz moléculas de água (H_2O). A cadeia resultante é chamada de cadeia peptídica e por meio dela se forma uma longa sequência de aminoácidos, sendo as extremidades inicial e final de uma cadeia peptídica quimicamente diferentes. A extremidade inicial carrega o grupo amino, sendo assim denominada terminal-N, enquanto a final carrega o grupo carboxila, sendo o terminal-C, portanto, uma proteína é sempre representada na forma da direção ‘N’ para ‘C’ (Branden e Tooze, 1999).

A Figura 2.2 representa uma ligação peptídica entre dois aminoácidos, podendo-se notar que o grupo α-amino de um desses (representado com o grupo R^2) atua como nucleófilo[4] para deslocar o grupo hidroxila de outro aminoácido (representado com grupo R^1), formando uma ligação peptídica, destacada pela região sombreada.

[4] espécie química que doa um par de elétrons para gerar uma ligação química durante uma reação (NELSON e COX, 2014).

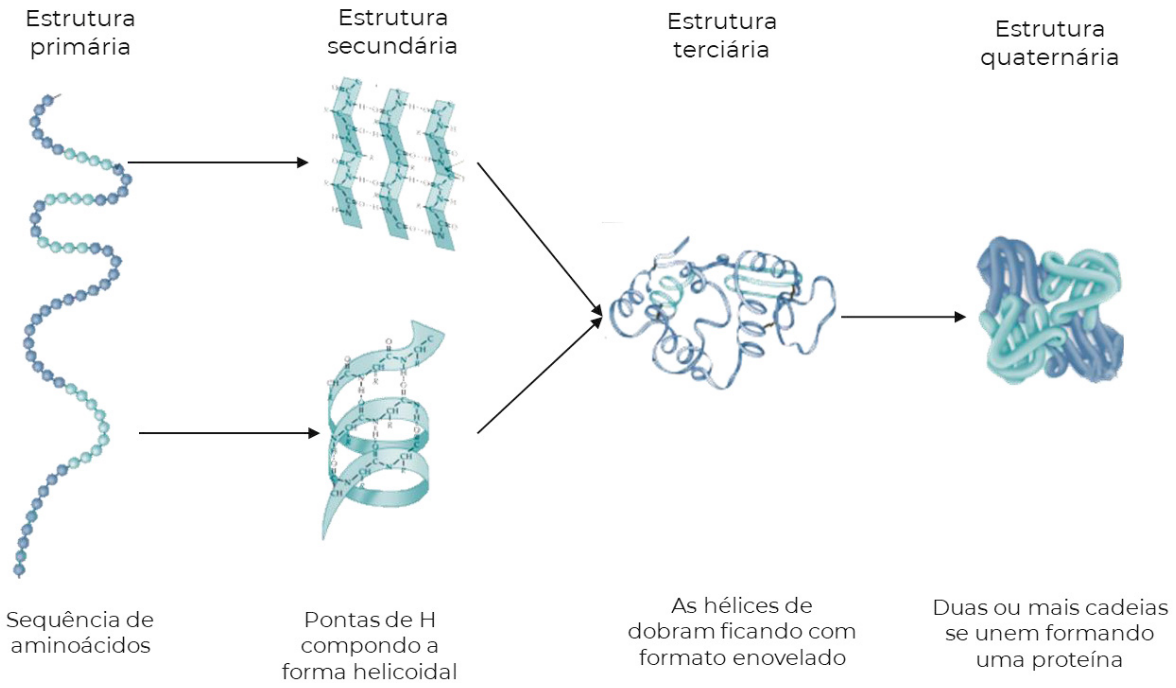
Figura 2.2 - Formação de uma ligação peptídica



Fonte: a autora, 2020.

Foram estipulados quatro níveis diferentes de complexidade de estruturas de proteínas, organizadas em uma espécie de hierarquia conceitual, as quais variam de uma sequência linear de aminoácidos até uma estrutura em três dimensões complexa, estas ilustradas na Figura 2.3. Resumidamente, a estrutura primária consiste em uma sequência de aminoácidos unidos por ligações peptídicas, chamada de polipeptídio, o qual pode ser disposto em unidades de estrutura secundária, como em uma hélice α. A hélice é uma parte da estrutura terciária do polipeptídeo dobrado, o que por sua vez é uma das subunidades que compõem a estrutura quaternária da proteína (Nelson e Cox, 2014).

Figura 2.3 - Níveis de estrutura nas proteínas



Fonte: a autora, 2020.

A estrutura primária, como já dito, é uma sequência linear de aminoácidos conectados por meio de ligações peptídicas. Essa estrutura não contém nenhuma informação geométrica. Contudo, uma vez que cada proteína possui uma quantidade e uma sequência de resíduos de aminoácidos diferentes, é possível descobrir outras informações interessantes a respeito da macromolécula. Da estrutura primária de uma proteína é proveniente a maneira como ela se dobra em sua estrutura tridimensional única, consequentemente determinando sua função (Nelson e Cox, 2014).

A estrutura secundária se refere a conjuntos particularmente estáveis de aminoácidos se organizando em padrões estruturais recorrentes. Esta pode ser considerada como a conformação local da cadeia polipeptídica, a qual é classificada entre dois tipos distintos: α-hélice, representada por uma cadeia de peptídeos em forma de haste enrolada formando uma estrutura helicoidal e, folha-β, caracterizada por duas fitas de peptídeos alinhadas na mesma direção (folha-β paralela) ou em direção oposta (folha-β antiparalela), sendo estável por ligações de hidrogênio (Jana *et al.*, 2018).

A estrutura terciária de uma proteína descreve as características do enovelamento tridimensional de um polipeptídeo, sendo assim, frequentemente

definida como a estrutura tridimensional global, a estrutura terciária é determinada como uma coleção de coordenadas 3D para cada átomo. As iterações das cadeias laterais dos aminoácidos determinam como será a estrutura terciária, isso ocorre quando a proteína se dobra de forma que suas partes hidrofóbicas se orientem para seu interior, deixando as partes hidrofílicas expostas à água e íons do ambiente (Scapin, 2005).

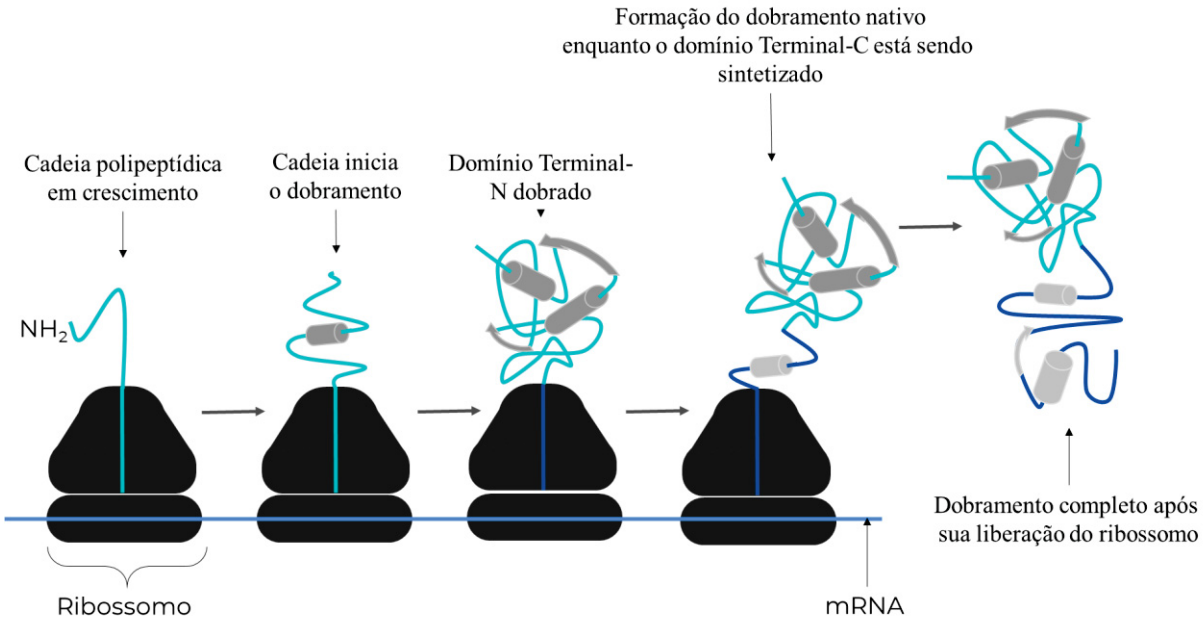
A estrutura quaternária refere-se ao arranjo espacial estável da união de duas ou mais subunidades polipeptídicas. Quando uma proteína tem duas ou mais subunidades polipeptídicas, seus arranjos no espaço são chamados de estrutura quaternária. Importante observar que nem todas as proteínas exibem estruturas quaternárias. As principais razões para a estabilidade das estruturas quaternárias são as mesmas iterações covalentes vistas nas estruturas terciárias, como por exemplo ligações de hidrogênio, interações de van der Waals e ligações iônicas (Jana *et al.*, 2018).

As sequências de proteínas foram selecionadas pelo processo evolutivo para atingir uma estrutura reproduzível e estável. As proteínas são a base para a vida, realizando processos indispensáveis para todos os organismos vivos. A compreensão molecular de como as proteínas funcionam é intrínseca no conhecimento e determinação das primeiras estruturas das proteínas, resultando no princípio biológico de que a função da proteína é assumida a partir da sua estrutura e o pré-requisito para gerar uma proteína funcional é um dobramento bem-sucedido e completo (Mészáros *et al.,* 2019).

2.1.1 Dobramentos de proteínas

A sintetização das proteínas é feita dentro de cada célula, as novas cadeias polipeptídicas gradualmente saem pela organela celular chamada de ribossomo, e assim, o dobramento, também chamado de enovelamento, já começa primeiramente no terminal N e progride vetorialmente antes mesmo que o terminal C esteja completamente insurgido do ribossomo (Waudby *et al*., 2019), conforme representado na Figura 2.4. Sucintamente, o dobramento é o procedimento pelo qual a informação linear contida na sequência de aminoácidos gerada dá origem à estrutura tridimensional da proteína funcional.

Figura 2.4 - Representação da sintetização de uma proteína



Fonte: a autora, 2020.

A conformação é associada à disposição espacial dos elementos químicos que formam uma proteína ou parte desta, ou seja, as possíveis conformações compreendem qualquer estado estrutural no qual proteína possa existir sem que suas ligações covalentes sejam quebradas (Nelson e Cox, 2014). Para que a proteína desempenhe sua função corretamente, essa organização dos átomos deve estar em seu mais alto grau com maior eficiência energética, essa conformação estrutural específica de cada proteína é chamada de conformação nativa, ativa ou natural (Yon, 2002).

Uma definição importante no dobramento de proteínas são os domínios, regiões da proteína os quais possuem funções e estruturas tridimensionais distintas e que podem se dobrar de forma autônoma. Como notado na figura acima, o Terminal-N se dobra individualmente antes do Terminal-C ser sintetizado, caracterizando assim, no caso da proteína representada na imagem, dois domínios distintos.

As proteínas tendem a se arranjar na conformação nativa naturalmente enquanto se formam em seus níveis estruturais discutidos na Seção 2.1, obedecendo às leis da termodinâmica, ficando a maior parte de sua vida em um estado de mínima energia livre. Assim, por ser a possível estrutura mais estável, a conformação nativa deve estar no estado global de mínima energia livre, sendo essa

a Hipótese da Termodinâmica criada por Anfinsen, Haber e White em 1957 após seus experimentos com proteínas *in vitro.*

Anfinsen (1973) verificou então, através de seus experimentos com proteínas tanto em ambiente *in vitro* quanto ao vivo, que as proteínas necessitam apenas das informações codificadas nas suas sequencias de aminoácidos para que a conformação com menor energia livre seja atingida, considerando também que é necessário que o domínio completo tenha sido formado para que se suceda o dobramento estável.

Alguns problemas conhecidos que geram a má formação de uma proteína dentro da célula são que a taxa de tradução do mRNA pelo ribossomo é menor que a taxa de dobramento, consequentemente, a proteína pode se dobrar incorretamente, danificar-se uma vez que não se encontra completamente dobrada ou também se juntar a outras cadeias peptídicas ao seu redor, gerando estruturas proteicas não funcionais (Waudby *et al*., 2019). Essas agregações de estruturas são mais perceptíveis em temperaturas mais altas e maiores concentrações de cadeias na célula.

Chaperones são moléculas cilíndricas que auxiliam para que agregações não ocorram, promovendo o dobramento das proteínas em seu interior. Elas são proteínas que se acoplam à estrutura instável que está sendo liberada pelo ribossomo e tem por objetivo estabilizá-las. Dessa forma, elas apenas auxiliam no dobramento correto, mas não são parte da estrutura funcional da proteína sintetizada (Scapin, 2005).

Uma vez que as proteínas formadas no ribossomo começam seu dobramento antes de estarem prontas, a energia mínima livre de uma parte do segmento da proteína é diferente da final quando a dobramento foi completado, para que isso não ocorra as chaperones se acoplam ao polipeptídio em síntese ainda até que todos os segmentos estejam prontos para que a cadeia se dobre funcionalmente e não prematuramente, isso ocorre de domínio a domínio (Schaffar *et al*., 2004). Além disso as chaperones também realizam outros processos relacionados às proteínas como transporte e até desdobramento e redobramento.

As chaperones possuem um trabalho essencial para prevenção de más formações em proteínas, contudo ainda existem casos que proteínas não dobradas ou dobradas incorretamente se acumulam (normalmente devido a calor excessivo), o que por consequência pode gerar mal funcionamento da célula como um todo,

uma vez que proteínas mal dobradas não somente perdem sua função original como podem inclusive simular uma função diferente e ainda prejudicar as células ao seu entorno (Schaffar *et al*., 2004).

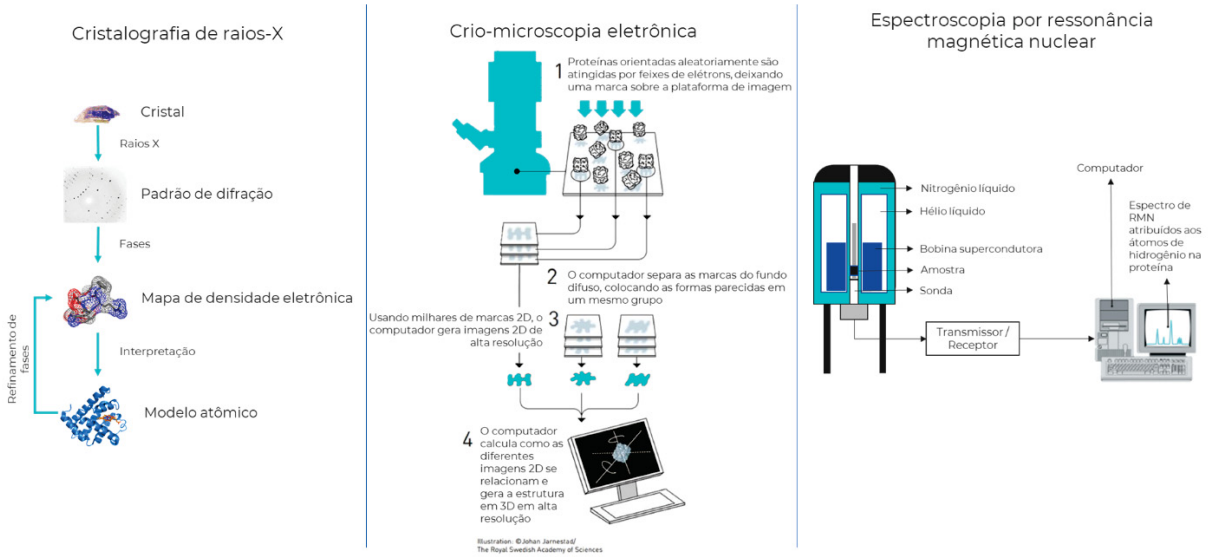
Consequentemente, algumas doenças conhecidas estão relacionadas a esses casos de má formação, como por exemplo doença de Alzheimer, Parkinson, alguns tipos de diabetes e cânceres, assim como outras doenças menos graves. Cada uma dessas doenças é associada a alguma proteína específica, porém o que elas têm em comum é uma grande quantidade de material proteico depositado interferindo nas funcionalidades celulares (Jana *et al.*, 2018).

2.2 PREDIÇÃO DA ESTRUTURA DE PROTEÍNAS

Como mencionado anteriormente, proteínas preferencialmente se dobram em uma estrutura tridimensional específica representada por sua conformação, processo assegurado pela capacidade de rotação das cadeias peptídicas. A estrutura final é chamada de estrutura nativa e consome o mínimo de energia livre possível (Jana *et al.*, 2018). O problema de estrutura de proteína se envolve em determinar a estrutura nativa de uma proteína, dada sua sequência de aminoácidos. A importância na biologia computacional desse tipo de previsão se dá na possibilidade de ser capaz de compreender as funções e o mecanismo de ação das estruturas proteicas.

Apesar dos avanços tecnológicos recentes destinados à determinação da estrutura de proteínas usando cristalografia de raios-X (Takaya *et al*., 2020), técnicas de espectroscopia por ressonância magnética nuclear (Park *et al*., 2020) e crio-microscopia eletrônica (Nygaard *et al*., 2020), mostrados na Figura 2.5, existe ainda uma crescente lacuna entre sequências e estruturas de proteínas conhecidas. Isso se deve ao fato desses métodos serem caros, uma vez que requerem equipamentos especializados, e demorados, como por exemplo o primeiro método que leva meses para ser completado e não existem regras sobre as condições ideais que resultam em um bom cristal de proteína.

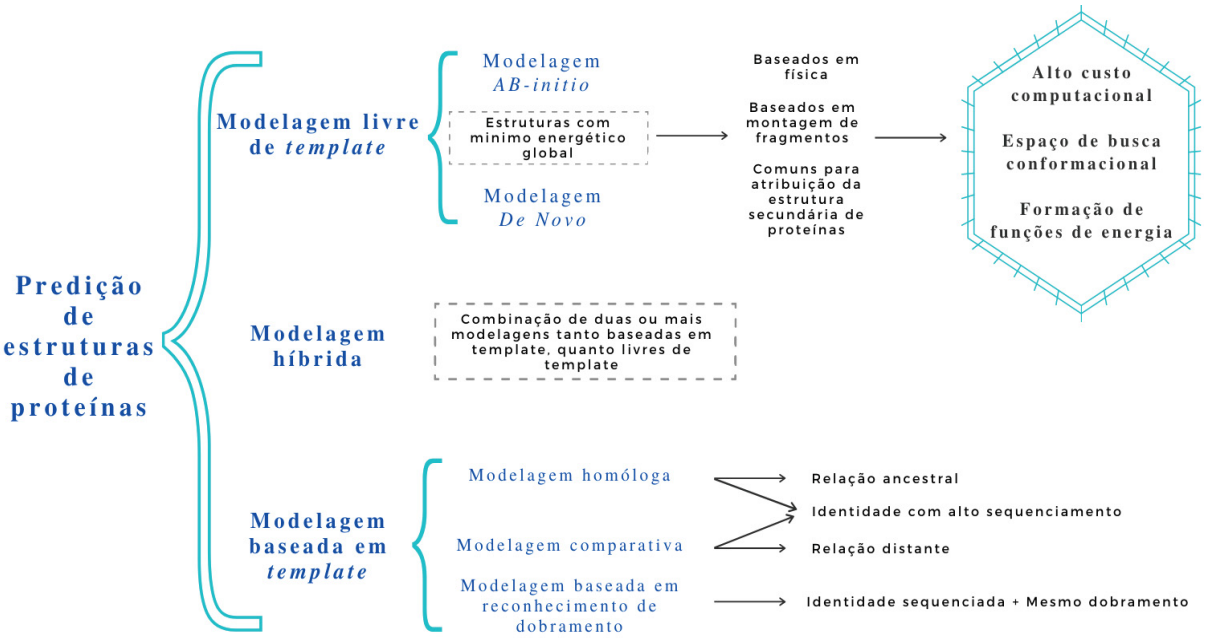
Figura 2.5 - Métodos laboratoriais de predição de estruturas de proteínas



Fonte: Academia Real Sueca de Ciências, 2017 (adaptado).

Essa necessidade de métodos mais ágeis e baratos para resolver o PSP criou a necessidade de metodologias de predição de estruturas de proteínas computacionais confiáveis (KAUSHIK *et al.*, 2018). Esses tipos de métodos qualificam-se pela utilização de recursos computacionais e não necessitam de atividades laboratoriais, demonstrando assim, potencial para superar as dificuldades associadas às abordagens experimentais. Os principais métodos computacionais para PSP são divididos em três categorias, as baseadas em *template*, ou seja, as que utilizam de base sequências de estruturas já conhecidas para comparar com as que serão preditas, as livres de *template* e as hibridas. Dhiangra *et al.* (2020) expõem que enquanto modelagens baseadas em modelos possuem grande sucesso na resolução do problema, as estratégias que envolvem modelagem livre de *template* ainda ficam para trás, principalmente quando se trata de proteínas consideradas grandes (mais de 150 aminoácidos). A Figura 2.6 apresenta a divisão dessas metodologias.

Figura 2.6 - Abordagens computacionais para a predição de estruturas de proteínas



Fonte: a autora, 2020.

A modelagem homologia ou modelagem homóloga (Ichinomiya *et al.*, 2020), a modelagem comparativa (Ginalski, 2006), e a modelagem por reconhecimento de dobramento (Yan *et al.*, 2021) são os tipos conhecidos de modelamento baseados em *template*. A primeira e a segunda são equivalentes, muitas vezes reconhecidas como sinônimos uma da outra e baseiam-se na comparação de similaridade de uma sequência de estrutura desconhecida (sequência alvo) com uma ou mais sequências de estruturas conhecidas (estruturas de modelo ou *template*), através de métodos de alinhamento sequencial (Guex e Peitsch, 2007), assim a precisão do método é determinada pela precisão do alinhamento, caso esta seja de 70% ou acima, as sequências alvo e modelo estão fortemente relacionadas e demonstram bons alinhamentos.

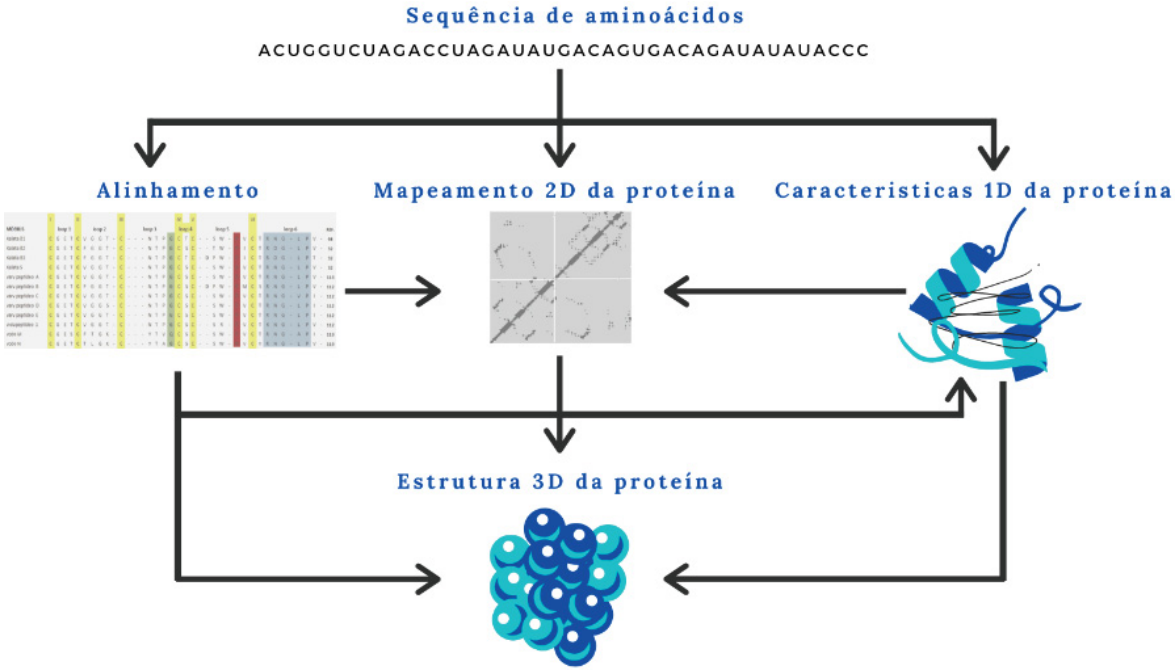
A terceira metodologia baseada em *template* é aplicado na previsão quando a sequência alvo tem pouca ou nenhuma semelhança de sequência primária com outra estrutura modelo conhecida, dessa forma esse processo é focado em encontrar semelhanças entre dobramentos e não sequências de aminoácidos (Jones, 1999). Entretanto, as abordagens *ab-initio* (Kaushik *et al.*, 2019) e *de novo* (Dawson *et al.*, 2019) são tipos de metodologias livres de *template*. Esses modelos

predizem a conformação tridimensional nativa de sequência primária de uma proteína se baseando nas propriedades hidrofóbicas e hidrofílicas dos aminoácidos.

O método *ab-initio* é fundamentalmente baseado na hipótese de termodinâmica Anfinsen, consequentemente, nessa metodologia a verificação de todas as conformações possíveis é necessária e a estrutura com a mínima energia é dita como estrutura nativa, uma vez que cada conformação é correspondida por uma energia. A abordagem de PSP *ab-initio* é equivalente a um problema de otimização global na qual a função de energia, representada pela função objetivo, deve ser minimizada. O principal problema com esta abordagem movida a energia é a explosão do tamanho do espaço de busca conformacional em função do comprimento da cadeia de uma proteína.

Uma solução para este problema consiste na exploração de abstrações mais simples, geralmente mais grosseiras, para orientar gradualmente a pesquisa, já que as proteínas parecem se dobrar localmente e não localmente ao mesmo tempo, mas formando formas mais complexas (Torrisi *et al*., 2020). Assim, a Figura 2.7 ilustra essa condição do dobramento de proteínas. Esse trabalho é focado em estudos com a abordagem *ab-initio*.

Figura 2.7 - *pipeline* generalizado de um PSP em *ab-initio*



Fonte: a autora, 2020.

É importante definir também a diferença entre o problema de predição da estrutura de proteínas e o problema de dobramento de proteínas. Sendo o primeiro voltado à predição da estrutura final da proteína, enquanto o segundo possui foco nos processos seguidos pela proteína até atingir seu estado nativo. No caso do PSP em *ab-initio* o segundo ajuda na realização do primeiro. Como mencionado, o problema de PSP em ab-*initio* é modelado como um problema de minimização da energia livre correspondente às possíveis conformações que uma proteína pode atingir.

Um modelo de dado que tenha esse princípio básico deve incluir um modelo dos átomos que formam a proteína demonstrando a sequência de aminoácidos e as ligações entre eles, um conjunto de regras que consegue descrever as possíveis conformações que a proteína pode assumir e uma função de energia que atribui a cada conformação um valor de energia livre (Pedersen, 2000).

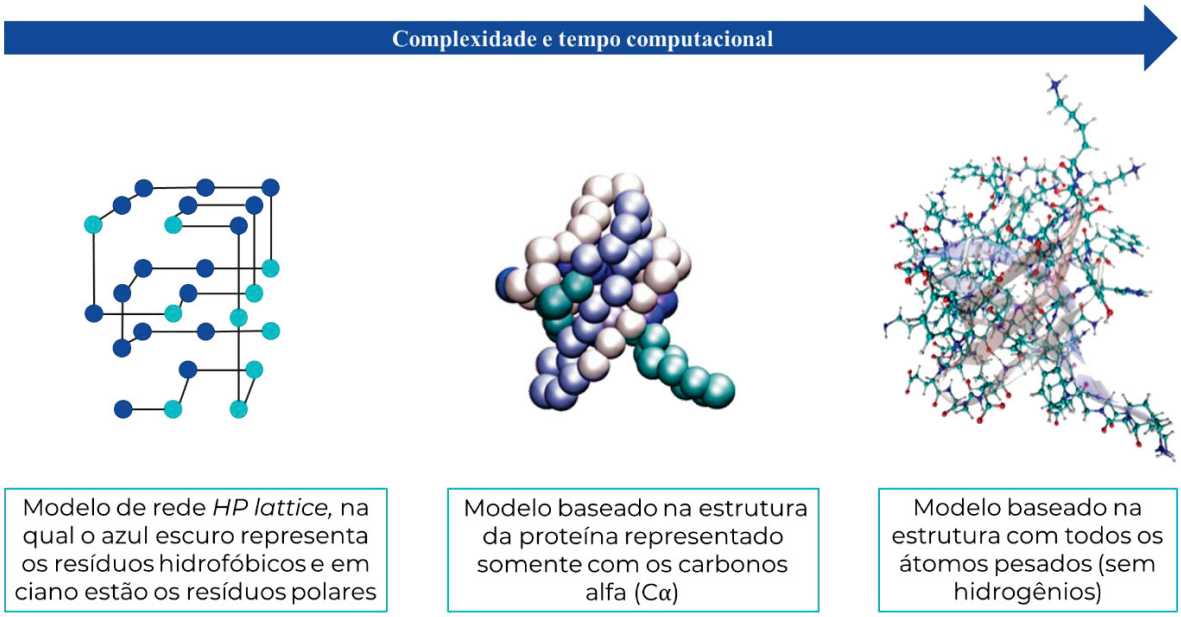
Para o modelo previsto ser considerado válido, ele deve possuir traços equivalentes da formação da estrutura de proteína real, como semelhança visual e estrutura tridimensional completa quando se trata por exemplo de uma predição de estrutura terciária, além de uma paridade entre o procedimento de formação da estrutura real e do modelo predito (Pedersen, 2000).

Devido à importância do dobramento de proteínas no funcionamento dos organismos, pesquisadores têm dedicado seus esforços ao entendimento de como esse processo realmente acontece, para que a conformação nativa de proteínas conhecidas possa ser determinada e, desta forma, também a sua funcionalidade.

2.3 MODELAGEM COMPUTACIONAL DE UMA PROTEÍNA

Ao utilizar abordagens computacionais para resolver o problema de PSP é necessário encontrar um modelo que possa representar a proteína abstratamente, demonstrando suas possíveis conformações e definindo sua função de energia livre a fim de avaliar tais conformações. As iterações atômicas da estrutura prevista dependem da escolha do modelo a ser seguido, além de que o detalhamento da estrutura predita depende da modelagem escolhida, fato expresso na Figura 2.8. O tipo de modelagem computacional simplificado mais utilizado são os modelos treliça (do inglês *lattice models*), que serão mais abordados nessa seção.

Figura 2.8 - Representação de uma proteína em diferentes abordagens computacionais



Fonte: Contessoto *et al.*, 2018 (adaptado).

O problema de PSP *ab-initio* pode ser abordado utilizando ambas as modelagens com ou sem treliça (Boiani e Parpinelli, 2020). Uma desvantagem notável dos modelos com treliça é que com tamanha simplicidade, alguns detalhes da representação da proteína real podem ser desconsiderados. Entretanto, esses modelos também não impõem restrições sobre os ângulos entre dois aminoácidos, permitindo representar computacionalmente proteínas com níveis de detalhe mais próximos das conformações reais.

2.3.1 Modelo HP Lattice

O modelo *HP lattice* foi proposto por Dill (1985) e é o modelo treliça discreto mais conhecido, simplificado e estudado na área de PSP. O HP é indubitavelmente o modelo *ab initio* de treliça menos complexo. Contudo, apesar de simples, esse modelo foi comprovado por Berger e Leighton (1998) como um problema NP-difícil (NP denota a complexidade polinomial não determinística). Dessa forma, a natureza NP-difícil do modelo HP é propagada para outras abstrações do PSP que tenham mais graus de liberdade (Boiani e Parpinelli, 2020).

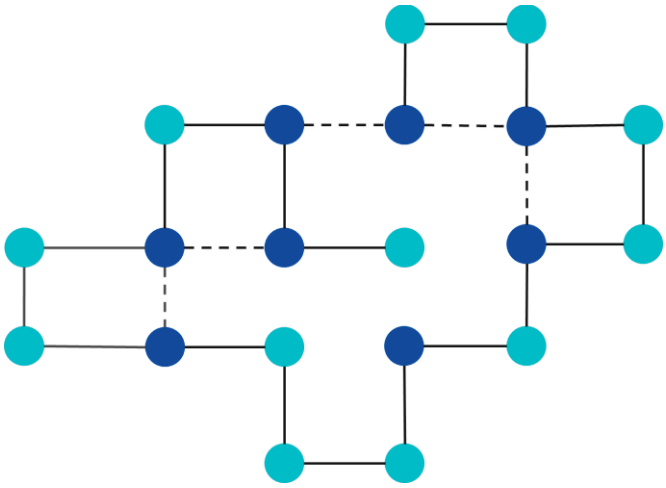
Esse modelo se fundamenta no pressuposto que a maior contribuição para o estado de energia livre de uma conformação nativa de uma proteína se deve ao

movimento dos aminoácidos hidrofóbicos em direção ao centro da proteína enquanto ficam envolvidos pelos aminoácidos hidrofílicos, os quais protegem o interior da proteína do solvente universal encontrado no ambiente.

Semanticamente falando, no que se refere a este modelo, H representa os resíduos hidrofóbicos ou não polares, enquanto P identifica aminoácidos polares ou hidrofílicos, sendo somente essas duas características que são identificadas em cada aminoácido. Dessa forma a proteína é representada por uma sequência binária {H, P} colocada em uma rede de forma que cada aminoácido ocupe uma posição da grade, onde aminoácidos sequenciais fiquem adjacentes uns aos outros. Essa grade quadrada pode ser de duas ou três dimensões, especificando modelos 2D ou 3D HP respectivamente. Esse modelo possui a restrição que dois ou mais aminoácidos não podem ocupar o mesmo ponto na grade, evitando sempre colisões.

A energia livre de uma conformação é inversamente proporcional ao número de ligações não locais, também conhecidas como ligações hidrofóbicas não locais ou ligações H – H não locais. Esse tipo de ligação refere-se ao encontro de um par de aminoácidos, ambos não polares, em posições adjacentes da rede, mas não adjacentes na sequência (sem ligações peptídicas entre eles). Consequentemente, a fim de minimizar a energia livre, procura-se maximizar o número de ligações não locais hidrofóbicas. A Figura 2.9 apresenta um exemplo de conformação com seis ligações H – H não locais, representados pelas linhas tracejadas, em uma grade 2D HP. Os pontos em azul escuro representam os aminoácidos hidrofóbicos, enquanto os cianos indicam os resíduos polares. No caso desse exemplo a energia livre dessa conformação tem valor de menos cinco (-5).

Figura 2.9 - Exemplo de ligações hidrofóbicas não locais no modelo 2D HP *lattice*.



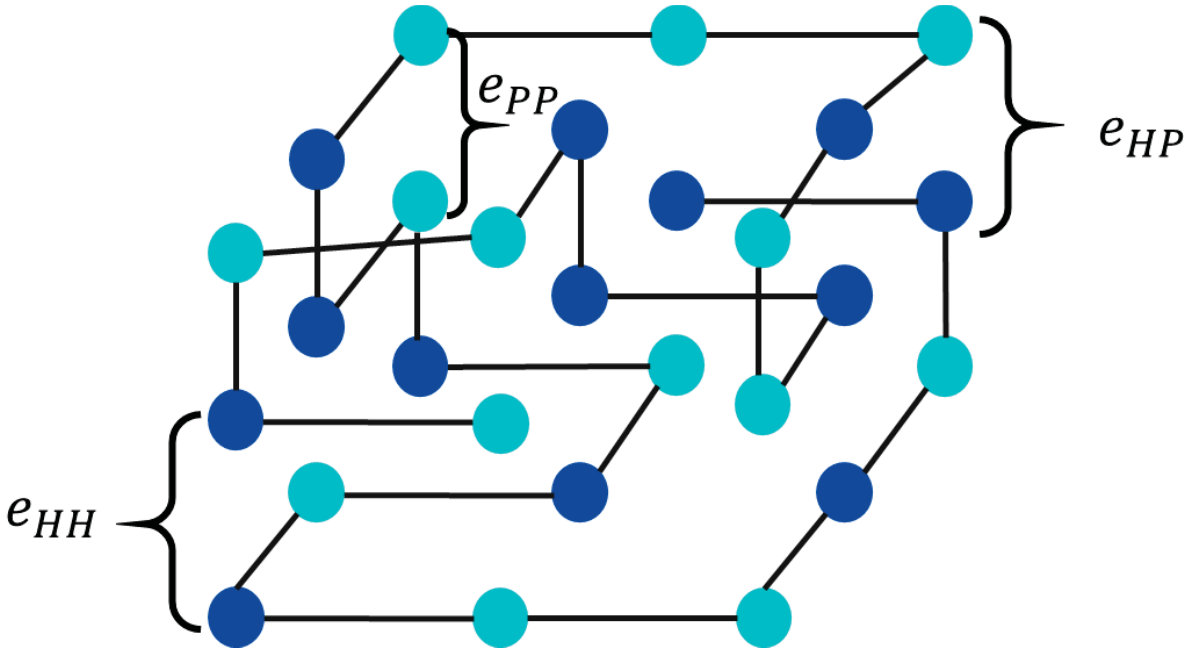
Fonte: a autora, 2020.

Helling *et al.* (1996) sugeriam a seguinte Equação 2.1 para demonstrar uma função simples de energia livre de uma conformação.

$$E = \sum_{i<j} e_{vi\,vj}\ \Delta(r_i - r_j) \qquad (2.1)$$

onde r_i e r_j representam os aminoácidos nas posições i e j na sequência, respectivamente; $\Delta(r_i - r_j)$ tem valor igual a um se os aminoácidos r_i e r_j possuem uma ligação não local, do contrário essa parte da equação se iguala a zero. Dependendo do tipo de contado entre os aminoácidos a energia e possui valores diferentes, correspondendo a menos um, caso o contato for de contatos H – H (caso e_{HH}), ou zero caso seja e_{HP} ou e_{PP}. Esses tipos de energia de contato são apresentados na Figura 2.10.

Figura 2.10 - Exemplo do modelo 3D HP apresentando os tipos de contatos.



Fonte: a autora, 2020.

O modelo HP *lattice* oferece uma facilidade para realização de simulações, pois, embora simples, o processo de dobramento no modelo age de maneira similar ao processo real. Além disso, grande parte das propriedades possuem comportamentos similares tanto na versão 2D quanto na 3D (Dill *et al.*, 1995).

2.3.2 Modelo AB Off-Lattice

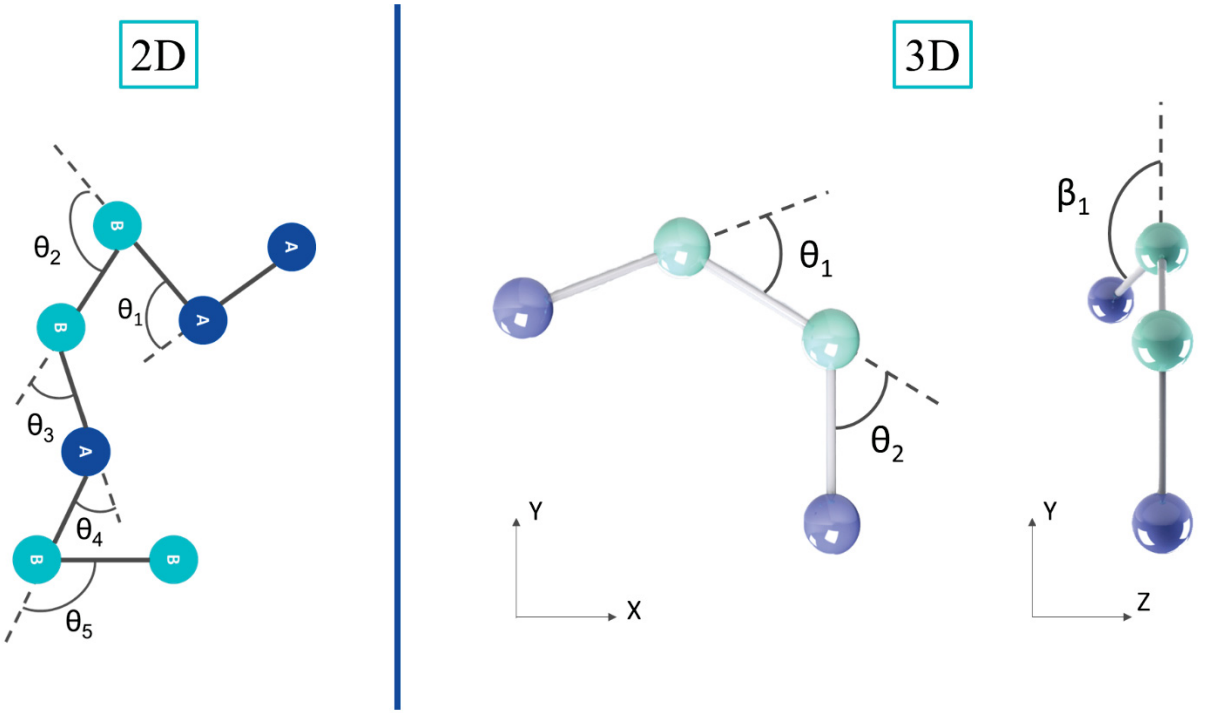
O Modelo AB *Off-Lattice* foi criado em 1993 por Stillinger *et al.* Assim como o modelo HP *Lattice*, essa modelagem classifica os vinte aminoácidos existentes entre resíduos hidrofóbicos ou hidrofílicos, demarcados pelas letras ‘A’ ou ‘B’, respectivamente. Esse modelo foi criado para predizer a estrutura de uma proteína a partir de sua estrutura primária e é muito utilizado devido ao fato de ser simples e conseguir caracterizar as interações intermoleculares entre os aminoácidos (Jana *et al.,* 2017).

Dessa forma, a sequência de aminoácidos é representada, analogamente ao HP *Lattice*, por meio de um vetor com sequência binária {A, B}, complementarmente são definidos também os ângulos de ligação ou curvamento entre os aminoácidos $\theta = \{\theta_1, \theta_2, ..., \theta_{n-2}\}$, onde n representa o comprimento da sequência. Esse ângulo de curvatura $\theta_i \in [-180^\circ, 0) \cup (0, 180^\circ]$ identifica se a rotação dos aminoácidos está no

sentido horário ou anti-horário, e caso seu valor se igualar a zero demonstra a linearidade nas ligações sucessivas. Quando se trata de uma representação 3D da proteína, inclui-se também os ângulos de torção $\beta = \{\beta_1, \beta_2, ..., \beta_{n-3}\}$ (BOŠKOVIĆ E BREST, 2020).

A Figura 2.11 mostra a posição de aminoácidos para uma sequência em duas dimensões e para uma em três dimensões, assim, a segunda metade da imagem mostra duas vistas da proteína 3D. Nota-se que a proteína em 3D é mais difícil de ser visualizada e criada pelo computador, por causa do maior número de dimensões e detalhes.

Figura 2.11 - Exemplo do modelo AB *Off-Lattice* apresentando os tipos de contatos.



Fonte: a autora, 2020.

Os aminoácidos ao longo da estrutura então são codificados por um conjunto de variáveis bipolares ξ_i. Caso o *i*-ésimo aminoácido for 'A', então ξ_i equivale ao valor um, já se o aminoácido for 'B', então ξ_i vale menos um. Utilizando esse princípio, uma função de energia potencial intramolecular Φ para qualquer sequência de proteína de comprimento *n* pode ser expressa usando a Equação 2.2.

$$\Phi = \sum_{i=2}^{n-1} V_1(\theta_i) + \sum_{i=1}^{n-2} \sum_{j=i+2}^{n} V_2(r_{ij}, \xi_i, \xi_j) \quad (2.2)$$

onde $V_1(\cdot)$ é a energia potencial de curvatura, definida pela Equação 2.3

$$V_1(\theta_i) = \frac{1}{4}(1 - cos(\theta_i)) \tag{2.3}$$

A iteração não conectada $V_2(\cdot)$ têm uma forma de Lennard-Jones (12, 6) dependente da espécie, descritas nas Equações 2.4 e 2.5, tal que

$$V_2(r_{ij}, \xi_i, \xi_j) = 4(r_{ij}^{-12} - C(\xi_i, \xi_j) r_{ij}^{-6}) \tag{2.4}$$

$$C(\xi_i, \xi_j) = \frac{1}{8}(1 + \xi_i + \xi_j + 5\xi_i\xi_j) \tag{2.5}$$

na qual r_{ij} representa a distância entre o i-ésimo e o j-ésimo resíduos da cadeia polipeptídica. Para um par AA, C (*ξi, ξj*) se iguala a 1, o que é considerado uma atração forte, já para um par AB ou BA, C (*ξi, ξj*) é igual a -0,5, considerado assim um repelente fraco e por último para um par BB, C (*ξi, ξj*) equivale a 0,5, sendo considerado como atração fraca entre os resíduos.

As coordenadas cartesianas para um resíduo i em um modelo AB *Off-Lattice* em três dimensões são obtidas pela Equação 2.6 onde

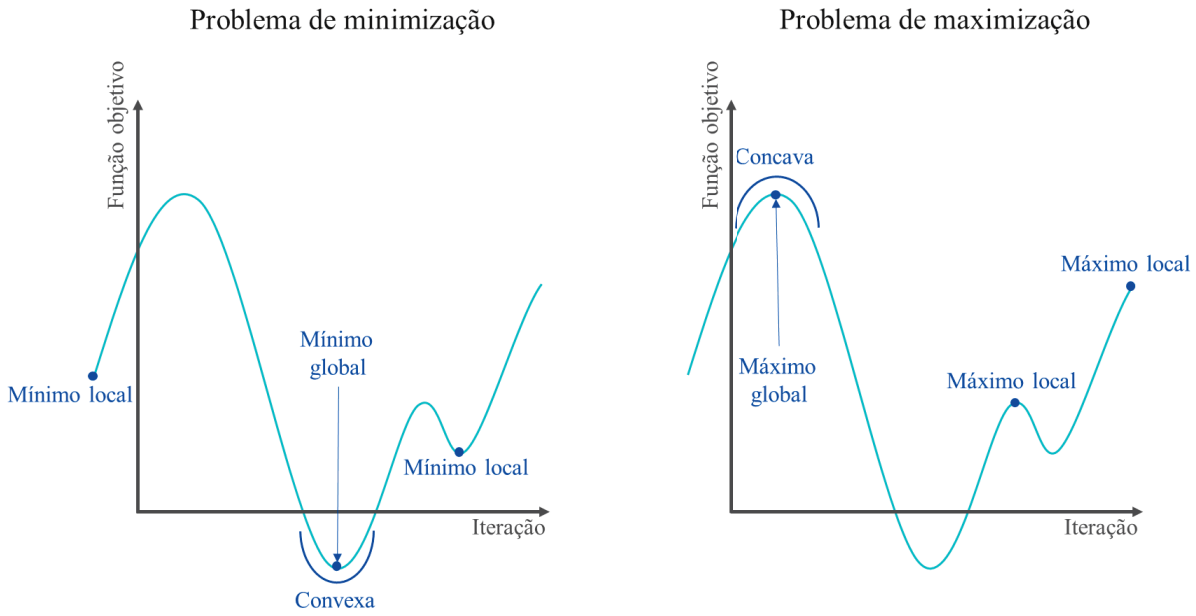
$$(x_i, y_i, z_i) = \begin{cases} (0,0,0), \text{ para } i = 1 \\ (0,1,0), \text{para } i = 2 \\ (\cos(\theta_1), sen(\theta_1) + 1, 0) \text{ para } i = 3 \; e, \\ (x_{i-1} + \cos(\theta_{i-2}) \times \cos(\beta_{i-3}), \\ y_{i-1} + \text{sen}(\theta_{i-1}) \times \cos(\beta_{i-3}) \\ z_{i-1} + \text{sen}(\beta_{i-3})), \text{para } 4 \leq i \leq n \end{cases} \tag{2.6}$$

Pode-se notar então que três primeiros resíduos são definidos no plano z = 0. Em seguida, a posição dos demais resíduos ($i \geq 4$) são calculados com base na posição do anterior. Em uma sequência de proteína, a energia proteica mínima (Φ) é obtida através de configurações ótimas dos ângulos (θ_1, θ_2, ..., θ_{n-2}, β_1, β_2, ..., β_{n-3}), associados à função de energia da proteína (Jana *et al.,* 2017).

2.4 METAHEURISTICAS DE OTIMIZAÇÃO

Um problema de otimização procura a solução mais adequada a partir de um conjunto de alternativas disponíveis dentro de restrições dadas, estabelecendo uma função objetivo. Uma solução viável que minimiza (ou maximiza) a função objetivo é descrita como a solução ótima. Um desafio encontrado nesse tipo de problema envolve o fato de que a menos que tanto a função objetivo quanto a região factível sejam convexas em um problema de minimização (analogamente, concavas em um problema de maximização), pode haver vários mínimos locais (máximos locais quando se trata de maximizar a solução), a Figura 2.12 ilustra esses casos. Um mínimo local possui uma solução pelo menos tão boa quanto a de qualquer ponto próximo, enquanto um mínimo global é pelo menos tão adequado quanto todos os pontos possíveis. Uma vez que a maioria dos problemas do mundo real são não convexos e NP-difíceis, as técnicas metaheurísticas acabam se tornando uma escolha popular para resolução dos problemas.

Figura 2.12 - Características de problemas de minimização e maximização



Fonte: a autora, 2020.

Heurística é um método que busca uma solução aproximada e não carece necessariamente de uma prova matemática de convergência, esse método é comumente utilizado nos problemas que não possuem uma boa resolução quando

utilizados os métodos tradicionais de otimização. Um método de metaheurística no contexto de solução de problemas de busca e otimização usa um conjunto de heurísticas, almejando encontrar uma solução quase ótima, ao invés de tentar encontrar especificamente a solução ótima exata, além de que geralmente não possuem prova rigorosa de convergência para a solução ótima e são computacionalmente mais rápidos do que os métodos convencionais (BANDARU E DEB, 2016).

As metaheurísticas utilizam de operações estocásticas na pesquisa de forma recorrente, possibilitando assim, modificar algumas das soluções candidatas iniciais, as quais são geradas aleatoriamente. Esse fato mostra a natureza iterativa das metaheurísticas. Além da potencialidade de oferecer uma melhor razão entre tempo computacional e qualidade da solução (Eberhart e Shi, 2001) quando comparadas aos métodos tradicionais, as metaheurísticas são mais flexíveis, uma vez que podem ser adaptadas para atenderem às necessidades do problema e não exigem nem que as restrições, nem a função objetivo sejam expressas tal como funções lineares das variáveis de decisão. Outras vantagens das metaheurísticas sobre os métodos clássicos de otimização incluem o fato de que essas podem levar a soluções boas o suficiente para os problemas NP-difíceis, elas também não necessitam de informações de gradiente e, portanto, podem ser usadas com funções objetivas não analíticas, de caixa preta ou baseadas em simulação, além de possuírem a capacidade de se recuperar de ótimos locais devido à estocasticidade inerente (BANDARU e DEB, 2016).

Dois conceitos importantes que determinam o comportamento de um método metaheurístico são a exploração e a *exploitation*. O primeiro se refere a quão bem os operadores diversificam soluções no espaço de busca, enquanto o segundo pode ser definido pela capacidade dos operadores de utilizar as informações disponíveis das soluções para intensificar a pesquisa. A exploração e a *exploitation* dão à metaheurística características de busca global e de busca local.

As metaheurísticas são usualmente baseadas em princípios naturais, físicos ou biológicos e tentam imitá-los em um nível fundamental por meio de vários operadores. As metaheurísticas usam o conhecimento sobre o problema para encontrar uma solução satisfatória em um tempo computacional razoável. Uma das vertentes das metaheurísticas são as baseadas em populações, comumente categorizadas em algoritmos evolutivos (do inglês *evolutionary algorithm*, EA) e as

baseadas em inteligência de enxame (do inglês *swarm intelligence*, SI) (Du e Swamy, 2016).

EAs se inspiram em aspectos da evolução na natureza, como a sobrevivência do mais apto, reprodução e mutação genética, os EAs mais conhecidos são algoritmos genéticos (do inglês *genetic algorithm, GA*), estratégias de evolução (do inglês e*volution strategies*, ES), evolução diferencial (do inglês *differential evolution*, DE), programação genética (do inglês *genetic programming*, GP) e programação evolutiva (do inglês *evolutionary programming*, EP). Já os algoritmos de inteligência de enxame imitam o comportamento de grupo e/ou interações de organismos vivos, como é o caso de formigas, abelhas, pássaros, vaga-lumes, peixes, glóbulos brancos e bactérias e, coisas não vivas também, exemplo são as gotas de água, sistemas fluviais e massas sob gravidade. As demais metaheurísticas que não se encaixam nas descrições anteriores, imitam fenômenos físicos como por exemplo *annealing* (Molina *et al.*, 2020) de metais e estética musical (harmonia).

As metodologias mais recorrentes para a resolução do problema de PSP, são as bioinspiradas e as inspiradas na natureza (Dhiangra *et al*., 2020). Ambas simulam diferentes processos biológicos naturais para abordar eficientemente problemas de otimização complexos. A segunda engloba a primeira, adicionando ainda algoritmos com inspirações em eventos físicos ou químicos (Molina *et al*., 2020).

2.4.1 Algoritmos de Inteligência de Enxame

Beni e Wang (1989) foram os primeiros a incluir o conceito de SI na área da robótica celular, contudo grupos de pesquisadores em diversas partes do globo começaram a pesquisar e estudar, quase que ao mesmo tempo, o comportamento de vários animais, em especial os insetos sociais foram a classe mais abordada. Foi esse comportamento coletivo de certas criaturas que motivou o princípio do que hoje é SI (Jana *et al.*, 2020).

A inteligência de enxame se refere ao comportamento coletivo de sistemas descentralizados e auto-organizados que são naturais ou artificiais. Normalmente, esses sistemas consistem em uma população de agentes simples interagindo localmente uns com os outros e com o ambiente. Os membros da população

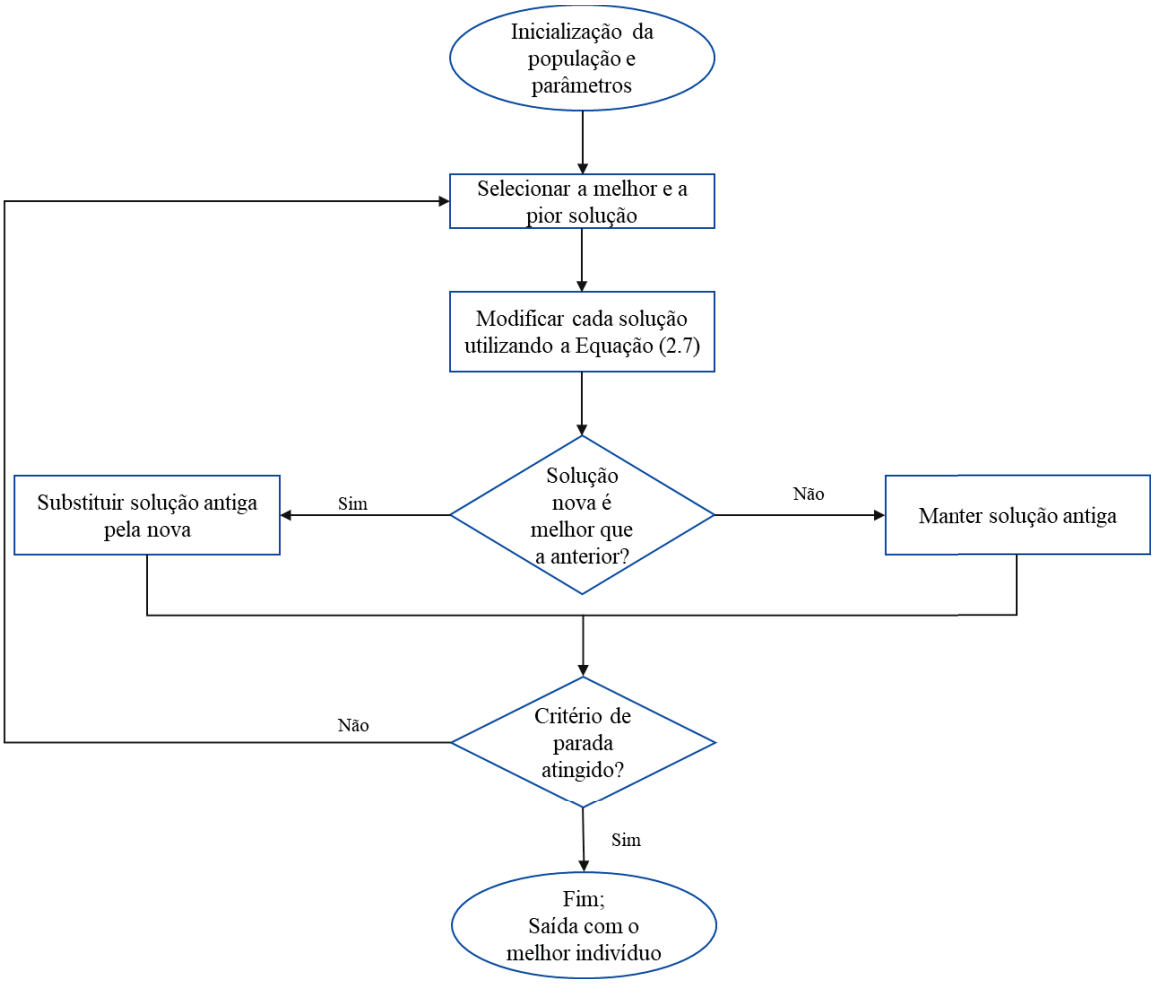
seguem regras simples e, mesmo que não haja uma estrutura de controle centralizado que dite como cada indivíduo deve se comportar, as interações locais quase aleatórias entre esses agentes levam ao surgimento de um comportamento global que pode ser considerado “inteligente” (Jana *et al.*, 2020).

2.4.1.1 Otimização Jaya

O algoritmo de otimização Jaya é conhecido por ser um algoritmo de busca heurística baseado em inteligência de enxame simples e mais rápido do que os demais (Pradhan e Bhende, 2019). Este é um algoritmo relativamente novo, criado em 2016 por Rao. Jaya significa “vitorioso” em indonésio, logo seu funcionamento se baseia na ideia de derrotar o adversário mais fraco e encontrar o caminho almejado até a vitória, dessa forma este algoritmo de otimização requer apenas que os parâmetros de controle comuns se movam em direção à melhor solução evitando a pior (Mishra e Ray, 2016).

Em um algoritmo Jaya multiobjetivo, a superioridade entre as soluções é decidida de acordo com o *rank* de não-dominância e o valor do parâmetro de densidade estimada, assim a solução com mais alta no *rank* e com maior valor de densidade é escolhida como melhor solução. Do contrário, a solução mais baixa no *rank* e com menor densidade é dita como sendo a pior solução. Esse tipo de modelo de seleção é adotado para que a solução na região menos populosa no espaço objetivo guie o processo de procura (Rao *et al*., 2017). A Figura 2.13 ilustra o fluxo dos dados em uma otimização Jaya.

Figura 2.13 - Fluxograma de uma otimização Jaya



Fonte: a autora, 2020.

Sequencialmente, as soluções são modificadas de acordo com a Equação (2.7). Na qual $X_{j,k,i}$ é o valor da *j*-ésima variável para o *k*-ésimo candidato durante a *i*-ésima iteração e $r_{1,j,i}$ e $r_{2,j,i}$ são pesos gerados aleatoriamente com distribuição uniforme no intervalo [0, 1] tal que

$$X'_{j,k,i} = X_{j,k,i} + r_{1,j,i}\left[X_{j,melhor,i} - \left|X_{j,k,i}\right|\right] - r_{2,j,i}\left[X_{j,pior,i} - \left|X_{j,k,i}\right|\right] \quad (2.7)$$

O valor de $X'_{j,k,i}$ é considerado satisfatório apenas se ele se provar melhor que $X_{j,k,i}$ na função *fitness*. Os valores mais adequados da função são armazenados no final de cada iteração e se tornam entrada para a próxima iteração, a fim de se obter no final a solução ótima (Pradhan, 2018). A Figura 2.14 representa um pseudocódigo para o algoritmo Jaya.

Figura 2.14 - Pseudocódigo de Jaya

Algoritmo 1 - Jaya

1: Inicializar tamanho da população número de *designed variables* e critério de parada
2: **enquanto** critério de para não for atingido **faça**
3: Verificar indivíduos para a melhor e a pior soluções
4: Modificar a solução com $X'_{j,k,i} = X_{j,k,i} + r_{1,j,i}[X_{j,melhor,i} - |X_{j,k,i}|] - r_{2,j,i}[X_{j,pior,i} - |X_{j,k,i}|]$
5: Atualizar a solução anterior se $X'_{j,k,i} > X_{j,k,i}$
6: **fim enquanto**
7: Mostrar a solução ótima estabelecida

Fonte: a autora, 2020.

A otimização Jaya é considerada viável para vários problemas de otimização por ser simples de converter seu código para qualquer linguagem de programação, sendo assim facilmente executável. A otimização Jaya, assim, é usada com êxito em diferentes aplicações como agrupamentos de documentos de texto (Thirumoorthy e Muneeswaran, 2021), otimização de parâmetros em um processo de eletroerosão (Gaikwad *et al*., 2021), controle de um motor de corrente contínua (Mohamed *et al*., 2020) e até em uma hidrelétrica (Chong *et al*., 2021).

Outra qualidade inclui o fato de que Jaya não precisa de nenhum parâmetro de algoritmo específico que precise ser ajustado antes do experimento computacional, ademais algoritmos que necessitam de parâmetros específicos tornam o experimento computacional de qualquer algoritmo demorado e, como tal, Jaya é eficiente em termos de tempo. Além disso, a natureza “vitoriosa” de Jaya faz com que o algoritmo escolha sempre a melhor solução (Alsajri *et al.*, 2018).

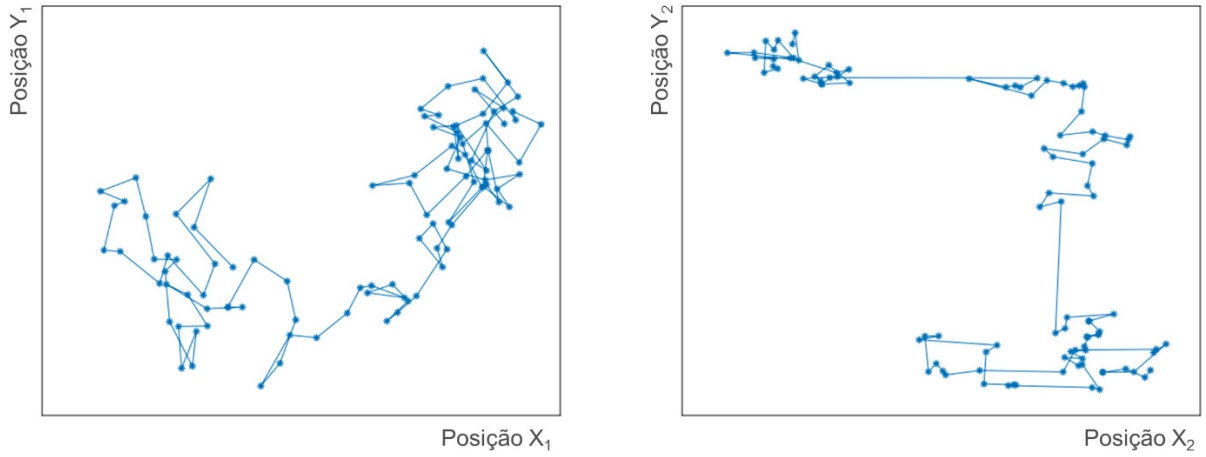
2.4.1.2 Variante de Jaya baseada em voo de Lévy

O voo de Lévy foi primeiramente constituído pelo matemático francês Paul Pierre Lévy em 1937 e caracteriza-se pela construção de trajetórias curtas e longas, sendo também classificado como um fractal de iteração aleatória. Voos de Lévy são uma classe especial passeios aleatórios (em inglês, *random walks*), ou seja, um processo estocástico no qual partículas ou ondas viajam ao longo de trajetórias aleatórias, nos quais os comprimentos dos passos durante a caminhada são

descritos por uma distribuição de probabilidade de “cauda pesada” (Barthelemy *et al.,* 2008). Sendo esta última uma distribuição que tende a zero mais devagar do que uma exponencial (Bryson, 1974).

Em um voo Lévy, as etapas do processo de caminhada aleatória seguem uma distribuição de lei de potência. Os voos da Lévy tornaram-se aplicáveis a uma ampla gama de campos, descrevendo padrões de caça de animais (Bartumeus *et al.,* 2005), distribuição de viagens de pessoas (Brockmann *et al.,* 2006) e até mesmo alguns aspectos do comportamento de terremotos (Corral, 2006). A Figura 2.15 compara um exemplo de um passeio aleatório com um voo de Lévy em duas dimensões, onde pode-se notar a diferença entre os dois. No primeiro ocorre aproximadamente o mesmo tamanho de passo para cada etapa, enquanto no voo Lévy há uma série de pequenos passos e ocasionalmente um grande passo.

Figura 2.15 – Exemplo de um passeio aleatório (esquerda) e um voo de Lévy (direita).



Autor: Iacca *et al.*, 2020, adaptado.

Nota-se que o padrão de movimento de um voo de Lévy ocorre de maneira que a partícula se move localmente primeiro, realizando uma série de pequenos passos, depois realiza um grande passo e volta então a se mover localmente (Iacca *et al*., 2020). Geralmente, a distribuição de Lévy é descrita em termos de transformada de Fourier e é expressa da forma

$$F(k) = \exp\left[-\alpha|k|^{\beta}\right], 0 < \beta < 2, \tag{2.8}$$

onde α é o parâmetro de assimetria ou fator de escala e se encontra dentro do intervalo [-1, 1], o sinal desse parâmetro denota a direção da inclinação, negativa para a esquerda e positivo para a direita e distribuição simétrica caso igual a zero. O índice de estabilidade β também é conhecido como índice de Lévy, o qual é um dos parâmetros mais importantes em um voo de Lévy, pois afeta substancialmente a distribuição, pequenos valores desse termo resultam em saltos longos, já e grandes valores de β causam saltos curtos (Ingle e Jatoth, 2020).

Conforme mostrado por Rao (2016), em alguns casos o algoritmo Jaya original pode não ser capaz de encontrar o ótimo global, devido à presença de ótimos locais que podem prender a busca. Nesse algoritmo, as soluções são modificadas considerando a melhor e a pior solução simultaneamente e, portanto, esta abordagem ajuda a acelerar a velocidade de convergência e aumentar a capacidade de exploração do algoritmo. No entanto, a diversidade da população e a capacidade de exploração do algoritmo podem ser enfraquecidas com uma rápida velocidade de convergência. Este problema resulta na convergência do algoritmo para um ótimo local.

Assim, Ingle e Jatoth (2019) propuseram a variante de Jaya com voo de Lévy (do inglês *Jaya algorithm with Lévy flight*, JayaLF), na qual o conceito de voo Lévy é incorporado ao manter a diversidade de soluções e, assim, aumentar a capacidade de explorar todo o espaço de busca. Além disso, o esquema de seleção ambicioso do algoritmo de DE é empregado para melhorar a capacidade de exploração sem perda da diversidade da população. E por último, um índice adaptativo de Lévy é introduzido a fim de manter o equilíbrio entre os recursos de exploração e aproveitamento do algoritmo ao longo do processo de busca.

Em 2020 Iacca *et al.* propuseram também uma variante de Jaya baseada no voo de Lévy (LJA), a qual será utilizada nessa pesquisa. Nela, ao invés dos números aleatórios r_1 e r_2 serem amostrados através de uma distribuição normal entre [0, 1], esses números serão encontrados utilizando a distribuição de Lévy, deixando o resto do algoritmo igual. Deve-se notar que o único parâmetro adicional em relação ao algoritmo de Jaya original é o índice da lei de potência β, necessário na Equação (2.8) para amostrar números aleatórios da distribuição de Lévy.

2.4.1.3 Variante de Jaya baseada em Oposição de Elite

A aprendizagem baseada na oposição de elite é uma técnica emergente no domínio de pesquisa heurística (Wang e Huang, 2018). Nela, as soluções originais e as soluções opostas são avaliadas e as melhores são mantidas para a próxima geração.

Para a otimização Jaya, dada a solução X_{melhor} com o melhor valor de fitness, a solução baseada em oposição de elite da solução individual X_k é definida como

$$X^*_{j,k,i} = r \cdot \left(da_j + db_j\right) - X_{j,melhor,i} \tag{2.9}$$

onde da_j e db_j são os limites dinâmicos do j-ésimo elemento, definidos por

$$da_j = \min\left(X_{j,k,i}\right) \tag{2.10}$$

$$db_j = \max\left(X_{j,k,i}\right) \tag{2.11}$$

Esses limites são atualizados a cada cinquenta gerações e a regra que é utilizada para garantir a solução dentro do limite é dada pela Equação (2.12)

$$X^*_{j,k,i} = rand\left(da_j, db_j\right), \text{se } X'_{j,k,i} < da_j \text{ ou } db_j > X^*_{j,k,i} \tag{2.12}$$

Assim, em um algoritmo Jaya baseado em Oposição de Elite (do inglês *Elite Opposition-Based Jaya Algorithm*, EO-Jaya), a melhor solução entre X e X^* é selecionada para a equação dada em (2.7). O EO-Jaya é mais bem demonstrado na Figura 2.16.

Figura 2.16 - Pseudocódigo para EO-Jaya

Algoritmo 1 - EO - Jaya

1: Inicializar tamanho da população (n) número de *designed variables* e critério de parada
2: **para** k:=1 até n **faça**
3: inicialize $X_{k,1}$
4: **fim para**
5: selecionar $X_{j,melhor,1}$ e $X_{j,pior,1}$
6: $i = 1$
7: **enquanto** critério de para não for atingido **faça**
8: **para** k:=1 até n **faça**
9: Gerar $X^*_{k,i} = r \cdot (da_j + db_j) - X_{melhor,i}$
10: **se** $X^*_{k,i}$ é melhor que $X_{k,i}$ **então**
11: Atualizar $X_{k,i} = X^*_{k,i}$
12: **fim se**
13: **para** j:=1 até d **faça**
14: Modificar a solução com $X'_{j,k,i} = X_{j,k,i} + r_{1,j,i}[X_{j,melhor,i} - |X_{j,k,i}|] - r_{2,j,i}[X_{j,pior,i} - |X_{j,k,i}|]$
15: **fim para**
14: **se** $X'_{k,i}$ é melhor que $X_{k,i}$ **então**
15: Atualizar $X_{k,i+1} = X'_{k,i}$
16: **senão**
17: Atualizar $X_{k,i+1} = X_{k,i}$
18: **fim se**
19: **fim para**
20: $i = i+1$
21: Atualizar $X_{melhor,i}$ e $X_{pior,i}$
22: **fim enquanto**

Fonte: a autora, 2021

2.5 REVISÃO DA LITERATURA

Há duas vertentes para realizar uma pesquisa literária, o mapeamento sistemático e a revisão sistemática da literatura. Essa última é considerada um meio de identificar, interpretar e avaliar estudos relevantes que abordam uma questão de pesquisa particular (Kitchenham e Charters, 2007). Sendo utilizada nessa seção, relacionando trabalhos desenvolvidos entre 1985 e abril de 2021 relativos às aplicações de metaheurísticas e aprendizado de máquina no problema de predição de estruturas de proteínas.

2.5.1 Revisão sistemática da Literatura

Esse capítulo abrange a utilização dos repositórios *Web of Science* (WoS) e Scopus para realizar um processo de coleta de dados e análise bibliométrica do estado da arte para o problema de PSP. A busca foi realizada no portal de periódicos da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES). O WoS é reconhecido por indexar publicações e periódicos de alta qualidade, além de conferências internacionais reconhecidas. Enquanto o Scopus é um repositório mais amplo, mesmo apresentando algumas inconsistências (SHUKLA *et al.*, 2019). Dessa forma, ambos bancos de dados são analisados e comparados, abrangendo publicações desde 1985 até maio de 2021.

Para gerar uma string de palavra-chave de busca pesquisada nas bases de dados utiliza-se de um operador AND (e) para conectar dois ou mais termos de busca e o operador OR (ou) para representar termos equivalentes, lembrando que cada base de dados possui sua própria sintaxe de busca. Como um dos critérios de inclusão é que a pesquisa deve ser realizada somente em artigos de revistas e conferências escritos em inglês, a *string* de busca genérica criada foi:

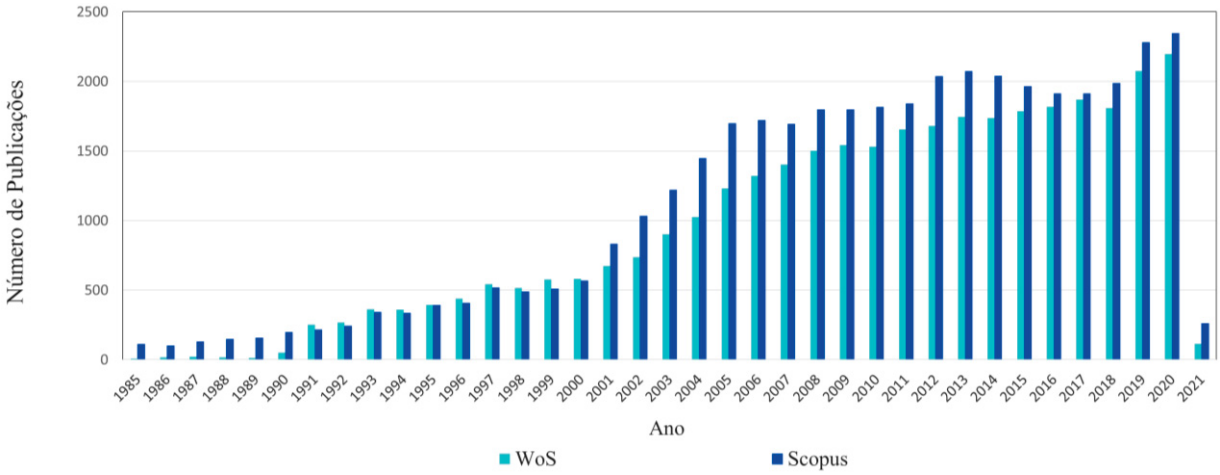
("Protein Structure prediction" OR *“PSP")* AND

("metaheuristic" OR *"differential evolution"* OR *“particle swarm optimization”* OR *"genetic algorithm")* AND

(“Machine Learning” OR *“ML”)*

O problema de PSP intriga cientistas há muitos anos, mas atualmente se tornou muito mais proeminente na comunidade cientifica devido ao crescente número de sequências encontradas anualmente, a compreensão de sua importância no controle das funções biológicas humanas e, é claro, na integração de métodos computacionais para a resolução do problema, o que aumentou as áreas de pesquisa não apenas às biológicas, mas também para engenharia e ciência da computação. O gráfico da Figura 2.17 mostra a quantidade de publicações por ano referentes a PSP, isso inclui pesquisas laboratoriais sem foco em automatização por processos computacionais.

Figura 2.17 - Número total de publicações ao longo dos anos

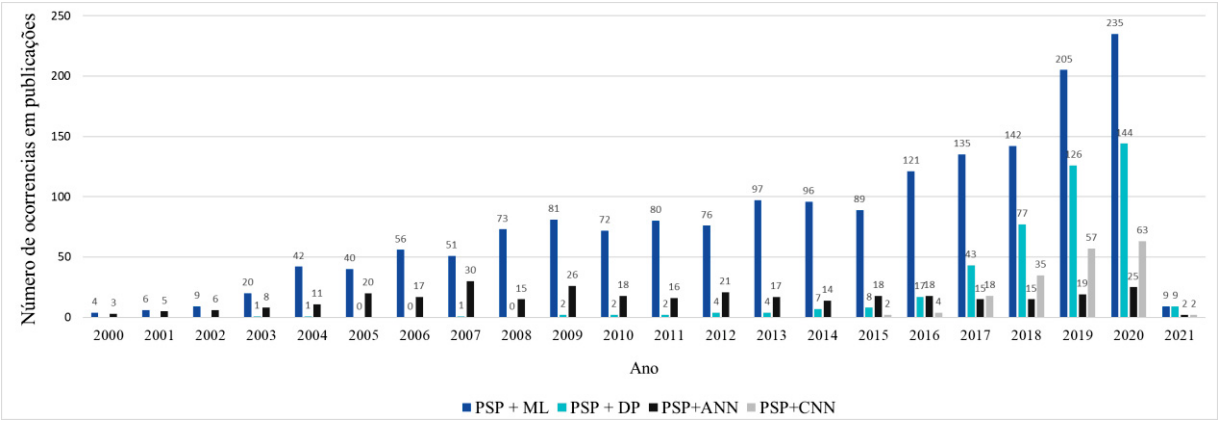


Fonte: a autora, 2021.

É possível notar que esta é uma área promissora, contudo grande parte da pesquisa ainda é em laboratórios especializados e o número de publicações existentes sobre o problema de PSP que utilizam apenas metodologias computacionais para solucioná-lo é apenas uma fração da quantidade total de pesquisas.

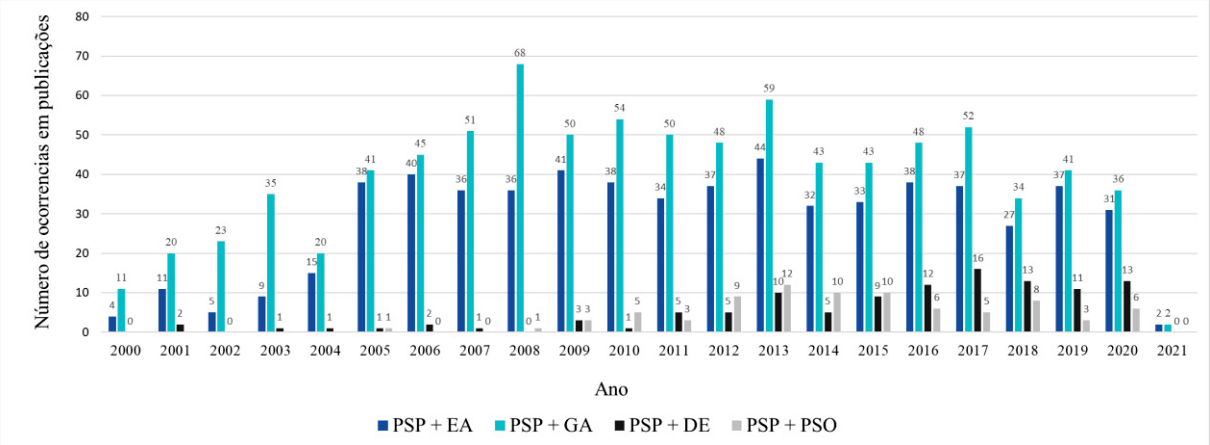
As Figuras 2.18 e 2.19 mostram a quantidade de publicações quando é especificado a metodologia computacional utilizada. Os dados foram coletados do repositório WoS, a partir somente do ano 2000, pois antes desse ano menos de dez publicações sobre os assuntos eram feitas por ano.

Figura 2.18 - Publicações por ano no WoS de metodologias de ML para o problema de PSP



Fonte: a autora, 2021.

Figura 2.19 - Publicações por ano no WoS de metodologias de metaheurísticas para o problema de PSP

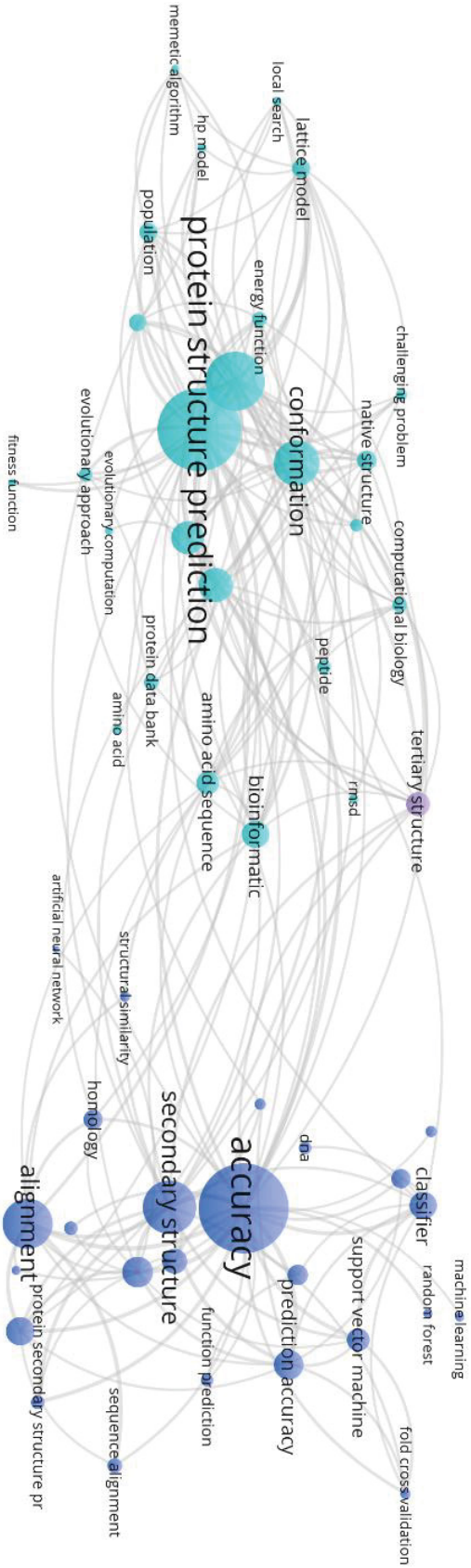


Fonte: a autora, 2021.

Pelos gráficos é possível notar que nos últimos anos a utilização de metodologias baseadas em aprendizado de máquina e aprendizado profundo é muito mais presente que a de metaheurísticas de otimização. Na primeira década do milênio houve um crescimento no uso de metaheurísticas, especialmente GA, que obtinham a mesma acurácia das técnicas de ML, fazendo com que o número de publicações sobre os assuntos fosse similar. Entretanto houve uma queda na utilização de algoritmos baseados em técnicas evolutivas no mesmo período que o uso de DL se tornou mais benéfico, uma vez que as técnicas foram melhoradas e a acurácia dos resultados cresceu consideravelmente quando comparada aos valores encontrados em metodologias de metaheurísticas na mesma época.

O VOSviewer é um *software* de visualização de informações amplamente utilizado para selecionar as melhores palavras-chave utilizadas pelos autores em seus artigos, assim, a Figura 2.20 mostra a interconexão dos conceitos mais presentes nas publicações do repositório WoS que englobem o problema de PSP e metodologias computacionais.

Figura 2.20 - Palavras-chave mais populares no WoS

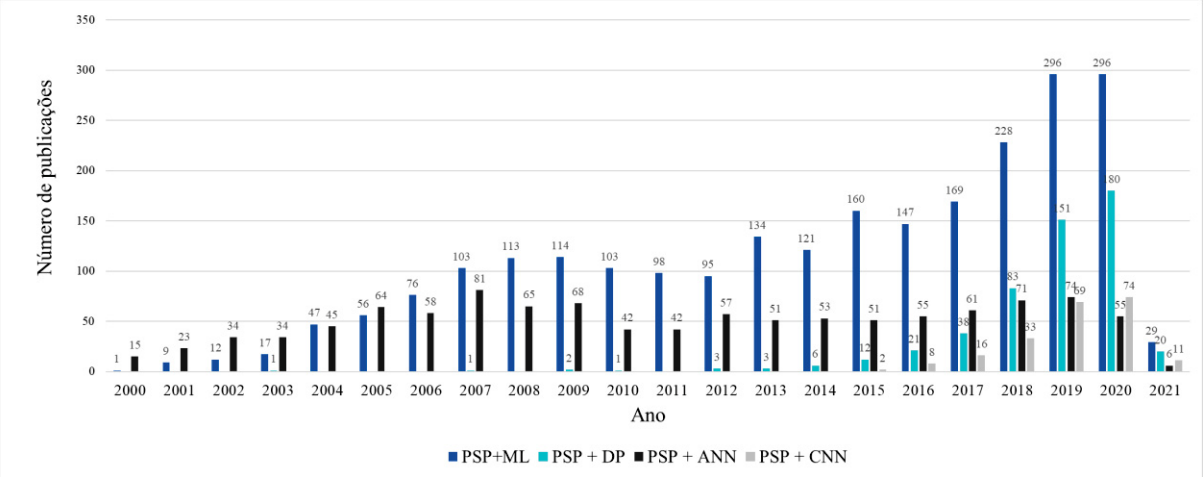


Fonte: a autora, 2021.

As palavras que mais cercam PSP incluem conceitos sobre as proteínas como aminoácidos, peptídeos e tipos de modelagens específicas. É interessante notar que “estrutura secundária” é muito utilizada, uma vez que grande parte dos artigos trabalhem com a predição apenas da estrutura secundária e não da estrutura quaternária 3D completa. Alguns métodos de aprendizado de máquina como floresta aleatória e máquina de vetor de suporte são presentes, assim como, é claro, redes neurais artificiais. Palavras que remetem a algoritmos evolutivos também são evidentes.

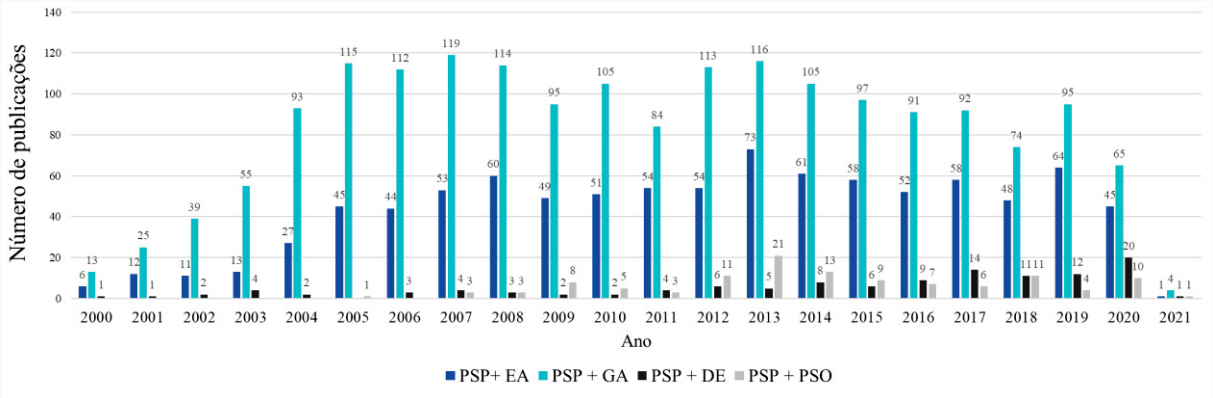
A mesma análise foi feita para o repositório Scopus. As Figuras 2.21 e 2.22 mostram a quantidade de publicações quando é especificado a metodologia computacional utilizada, e assim como o outro repositório foram analisados apenas publicações do período de 2000 até o maio de 2021.

Figura 2.21 - Publicações por ano no Scopus de metodologias de ML para o problema de PSP



Fonte: a autora, 2021.

Figura 2.22 - Publicações por ano no Scopus de metodologias de metaheurísticas para o problema de PSP

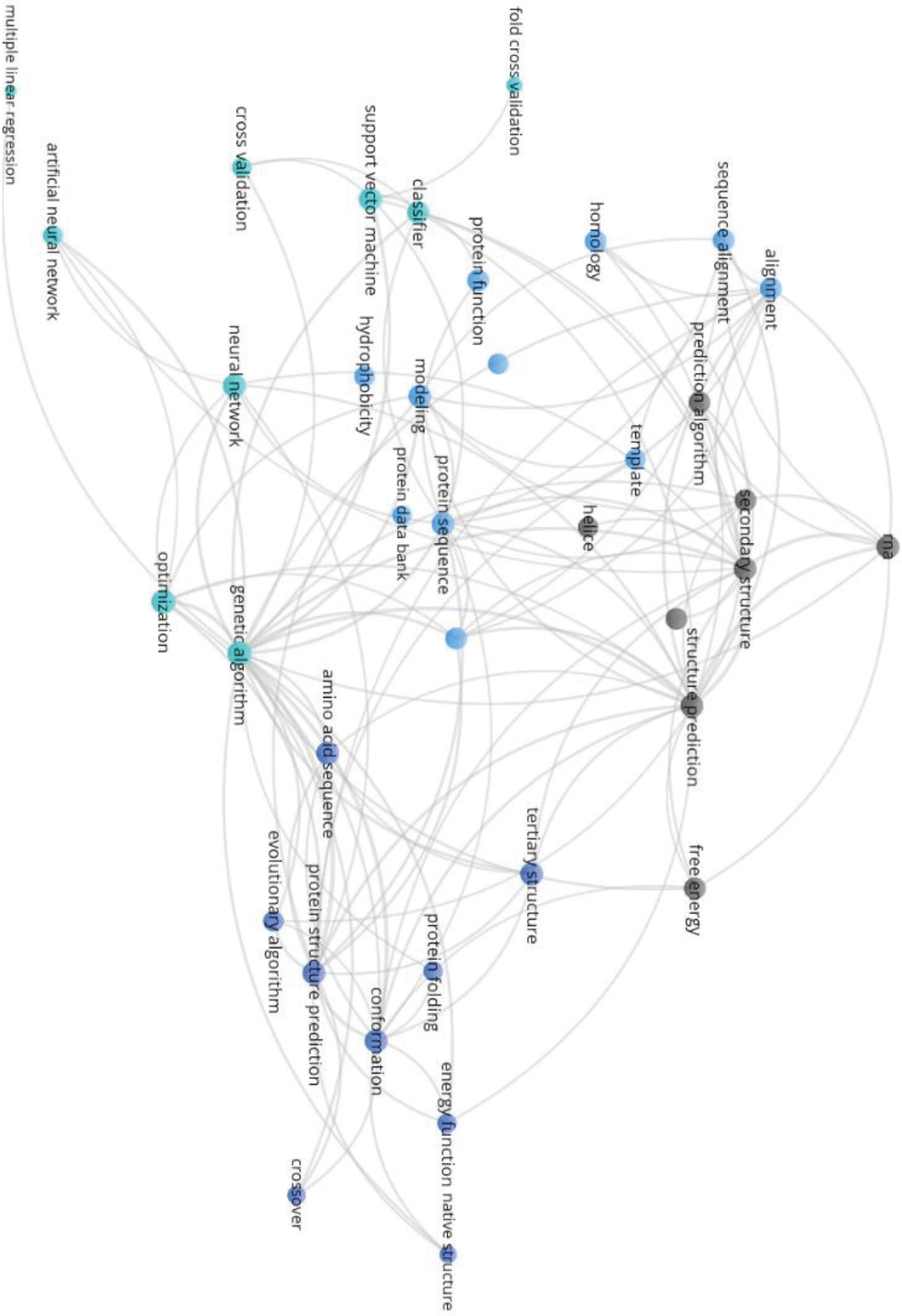


Fonte: a autora, 2021.

O número de publicações no Scopus segue a mesma distribuição que no WoS, apenas com a visível diferença de que no Scopus o número de artigos que incluem o uso de GA no problema de PSP é praticamente o dobro do que no outro repositório, mostrando como esse é um algoritmo bastante comum para essa aplicação, ao contrário da Evolução Diferencial com no máximo onze artigos publicados ao ano usando-a.

A Figura 2.23 foi gerada também com o software VOSviewer e mostra a interconexão das palavras chaves mais utilizadas nas publicações envolvendo PSP e modelagem computacional encontradas no Scopus.

Figura 2.23 - Palavras-chave mais populares no Scopus



Fonte: a autora, 2021.

A maioria das palavras chaves encontradas nas publicações provindas do WoS se repetem nas do Scopus, mostrando a consistência dos artigos. A maior diferença está apenas na frequência que as palavras são vistas, tendo o Scopus menos discrepância no número de ocorrências de cada termo.

As tabelas 2.2 e 2.3 mostram as áreas de pesquisa mais recorrentes em relação a artigos publicados com o tema de uso de metaheurística para o problema de PSP. A primeira é referente ao WoS e a segunda ao Scopus.

Tabela 2.2 Áreas do conhecimento mais frequentes relacionadas com o uso de metaheurísticas no problema de PSP de acordo com WoS

Área do conhecimento	Frequência Absoluta	Frequência Percentual
Biologia Matemática Computacional	13	15,66%
Ciência da Computação	12	14,46%
Matemática	12	14,46%
Biologia Molecular e Bioquímica	8	9,64%
Biofísica	7	8,43%
Engenharia	5	6,02%
Hereditariedade Genética	4	4,41%
Biomedicina e Ciências da Vida	4	4,41%
Controle e Automação de Sistemas	3	3,61%
Outros	25	30,12%

Fonte: a autora, 2021.

Tabela 2.3 Áreas do conhecimento mais frequentes relacionadas com o uso de metaheurísticas no problema de acordo com Scopus

Área do conhecimento	Frequência Absoluta	Frequência Percentual
Ciência da Computação	48	28,23%
Matemática	40	23,53%
Bioquímica, Genética e Biologia Molecular	30	17,64%
Engenharia	13	7,65%
Ciências Biológicas e Agricultura	7	4,12%
Medicina	5	2,94%
Química	5	2,94%
Profissões da Saúde	3	1,76%
Multidisciplinares	3	1,76%
Outros	16	9,41%

Fonte: a autora, 2021.

Nota-se que o WoS apresenta maior número de áreas do conhecimento que foram relacionadas ao uso de metaheurísticas no problema de PSP, por isso os números absolutos são menores do que quando comparado ao Spocus. Contudo, em áreas iguais há uma porcentagem de aparição similar nos dois diretórios, como é o caso da área de “Engenharia” que apresentou uma frequência de seis e sete porcento no WoS e no Spocus, respectivamente.

As Tabelas 2.4 e 2.5 apresentam as revistas e congressos com duas ou mais publicações que utilizaram metaheurísticas no problema de PSP, de acordo com o que foi pesquisado no diretório WoS e Scopus, respectivamente.

Tabela 2.4 Revistas e congressos mais frequentes relacionadas com o uso de metaheurísticas no problema de PSP de acordo com WoS

Revista/Congresso	**Frequência Absoluta**	**Frequência Percentual**
IEEE Transactions on Evolutionary Computation	3	10,34%
2013 IEEE Congress on Evolutionary Computation	2	6,90%
Plos One	2	6,90%
Advances In Swarm Intelligence ICSI 2016	2	6,90%
2016 19TH International Conference on Computer and Information Technology	2	6,90%
Ain Shams Engineering Journal	2	6,90%
Outros (uma cada)	16	55,17%

Fonte: a autora, 2021.

Tabela 2.5 Revistas e congressos mais frequentes relacionadas com o uso de metaheurísticas no problema de acordo com Scopus

Revista/Congresso	**Frequência Absoluta**	**Frequência Percentual**
Lecture Notes in Computer Science Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics	5	6,67%
Computational Biology and Chemistry	4	5,33%
BMC Bioinformatics	3	4,00%
Journal Of Bioinformatics and Computational Biology	3	4,00%
Plos One	3	4,00%
Swarm And Evolutionary Computation	3	4,00%

Bioinformatics	2	2,67%
Outros (uma cada)	52	69,33%

Fonte: a autora, 2021.

Enquanto o WoS apresenta mais congressos em seu banco de dados, o Scopus mostra uma relevância maior em revistas. Além disso a quantidade de publicações em cada revista ou congresso se mantém constante, sendo que a maioria tem apenas uma publicação sobre o assunto, e as que tem mais (observadas nas tabelas) não ultrapassam uma quantidade de cinco publicações.

2.5.2 Síntese dos trabalhos mais citados sobre PSP

Utilizando os princípios de EA, Parpinelli e Lopes (2013) utilizaram um algoritmo inspirado na ecologia para resolver o problema, o qual aplicava estratégias de busca cooperativa onde a população de indivíduos evoluía e interagia entre si usando conceitos ecológicos. Olson *et al*. (2013) também desenvolveram um EA baseado em *crossover* homólogo para resolver o problema de PSP. Os autores incorporaram recursos *ab-initio* para melhorar o estado da arte da época no método de previsão de estrutura.

Utilizando outra vertente, Kalegari e Lopes (2013) propuseram duas versões de DEs, a básica e a adaptativa, para resolver o problema de PSP. Boškovic e Brest (2016) também propuseram uma variante de DE, que era auto adaptativa para melhorar a eficiência e reduzir o número de parâmetros de controle para otimização do dobramento de proteínas.

Em 2014, Sar e Acharyya implementaram seis variantes de GAs, as quais são caracterizadas pelo uso de diferentes técnicas de seleção e *crossover*. As variantes propostas são empregadas em técnicas de seleção aleatória, elitista e de torneio e passam por dois tipos de recombinações.

Outro método de otimização mencionado na literatura desde 2010 são as metaheurísticas baseadas em inteligência de enxames (do inglês *swarm intelligence*). Cheng-Yuan *et al*. (2010) desenvolveram uma PSO com comportamento quântico, no qual a população é dividida em subpopulações de elite, exploração e *exploiting* para determinar a estrutura da proteína. Anteriormente, Zhang e Li (2007) propuseram esse mesmo tipo de arquitetura, na qual a

subpopulação elitista é obtida usando uma estratégia de mutação gaussiana. As posições de todas as partículas são atualizadas por meio do algoritmo PSO para subpopulação exploradora. Finalmente, as posições das partículas no subgrupo exploratório são geradas usando uma estratégia de exploração. Assim, o esquema proposto ajuda a escapar dos mínimos locais e fornece maior velocidade de convergência.

Apesar de complexo, esse método teve de ser bastante aperfeiçoado ao longo dos anos, uma vez que os experimentos só obtinham uma boa porcentagem de sucesso em entidades de proteínas artificiais de comprimento entre treze e 55 aminoácidos. Atualmente a maioria dos algoritmos consegue decodificar estruturas de proteínas de até 150 aminoácidos, tendo ainda alguns que computem mais de 200, porém com menor taxa assertiva. Esses algoritmos são normalmente híbridos de duas ou mais técnicas

Dhiangra *et al*. (2020) discutem que apenas bem recentemente, as abordagens de aprendizagem profunda baseadas em redes neurais se tornaram grandes na área da previsão da estrutura de proteínas. Contudo, as abordagens de aprendizado profundo para o problema de PSP têm sido amplamente utilizadas apenas como um dos componentes de todo o algoritmo, em vez de implicitamente implementadas como a principal ferramenta.

Em 2019, Al-Quraishi focou na construção de uma abordagem de previsão baseada totalmente em aprendizado profundo. O algoritmo baseou-se em dados derivados apenas da sequência da proteína em questão e do perfil evolutivo de resíduos individuais dentro da sequência. Este método obteve resultados bons, em comparação à outras técnicas mais consolidadas, conforme observado em caso de modelagem *ab-initio*.

Dhiangra também ilustra que a exploração dos benefícios das abordagens de aprendizado profundo ainda está começando no problema de PSP. Ainda é comentado que a principal dificuldade que pode ser encontrado por essas técnicas estaria relacionado à falta de disponibilidade de dados estruturais, uma vez que essas abordagens são baseadas no treinamento de algoritmos com base em certos padrões seguidos pelos dados disponíveis.

A área relacionada à previsão de estrutura, como descrito no capítulo 1, sempre foi mais lenta do que o equivalente de sequenciamento, o qual consequentemente dispõe de menos disponibilidade de dados que podem ser

usados para treinar um algoritmo. Outro problema que essas abordagens podem apresentar é o treinamento excessivo dos algoritmos.

Torrisi *et al*. (2020), por outro lado, pesquisa na utilidade das técnicas de aprendizado profundo para passos intermediários de previsão, no qual abstrações são inferidas, que são mais simples que a estrutura proteica completa, isto é chamado de anotações de estruturas de proteínas (AEP). É exaltado o uso de CNN nesse tipo de caso.

Os principais desafios enfrentados na resolução do problema de PSP estão em encontrar uma maneira eficiente de explorar o espaço de busca conformacional, uma vez que uma sequência de proteína pode dobrar em formas indefinidas. Assim, reduzir as possibilidades de dobra plausíveis para a melhor dobra provável é uma tarefa difícil de alcançar. A melhor estratégia de design desenvolvida na literatura refere-se à limitação do comprimento da sequência.

As previsões encontradas na literatura científica com maior porcentagem de confiabilidade são as que preveem estruturas proteicas menores que o comprimento de 150 aminoácidos, embora também seja visto esse limite se expandir até um comprimento de 200 a 250 aminoácidos. A única exceção desse caso observada nos últimos anos de competição CASP[5] foi no trabalho de Moult *et al*. (2016), no qual uma proteína de comprimento 256 aminoácidos foi prevista com precisão.

Em contra partida, o trabalho de Rives *et al*. (2020) concentra-se no uso de técnicas baseadas em programação de linguagem natural e aprendizado profundo não supervisionada em sequências de aminoácidos. O algoritmo aprendeu uma representação das sequências que codificam várias propriedades das proteínas, como a estrutura 3D e as relações evolutivas. Esse modelo, que foi disponibilizado ao público, contém aproximadamente 700 milhões de parâmetros, foi treinado em 250 milhões de sequências de proteínas e aprendeu representações de propriedades biológicas que podem ser usadas para melhorar o estado da arte atual em várias tarefas de predição genômica.

Jana *et al.* (2017) citam alguns problemas encontrados na resolução do problema de PSP. Entre eles, é mencionado o fato de mesmo utilizando computadores potentes para calcular modelos de todos os átomos do processo de

---

[5] Sigla em inglês para Avaliação Crítica da Previsão de Estrutura de Proteínas, experimento mundial para a previsão de estruturas de proteínas em toda a comunidade científica. Essa competição ocorre a cada dois anos desde 1994.

dobramento de proteínas, ainda assim eles são úteis apenas para um determinado horizonte de tempo ou comprimento de proteína. Além disso, os autores aludem que é possível encontrar sequências de proteínas em informações de estrutura conhecida por meio de um processo de modelagem de homologia para ver quão bem uma nova sequência é mapeada para uma estrutura conhecida, contudo isso não é considerado um dobramento *ab-initio*.

Sobre a abordagem de metaheurísticas, Jana *et al.* (2017) relembram que é necessário o conhecimento de uma função objetiva certa para a proteína e seu ambiente, o que pode em alguns casos ser difícil de definir. Assim, concluem que à medida que o conhecimento sobre sequências, estruturas e seu mapeamento melhora, nota-se que as proteínas dos sistemas vivos estão em constante movimento, consequentemente, a incapacidade de recapitular fielmente a estrutura correta pode ser um componente da imprecisão do processo real ou da incapacidade de capturar a função de adequação de uma maneira que possibilite o sucesso de abordagens baseadas na física ou metaheurísticas.

Song *et al*. (2018) propõem em seu artigo um método de resolução do problema de PSP utilizando otimização multiobjetivo (MOOP, do inglês *multi-objective optimization problem*) baseada em PSO. Combinando uma função energética baseada em física e uma função energética baseada no conhecimento para construir uma função de energia de três objetivos, diferenciando da maior parte da literatura que tende por usar apenas uma dessas funções sem combinações. A minoria que utiliza MOOP em suas pesquisas costuma dar mais atenção à função energética baseada em física e os autores advertem sobre isso, uma vez que afirmam que as funções baseadas em conhecimento tendem a ser mais efetivas.

Para resolver um MOOP a solução precisa otimizar mais de uma função objetivo, o que é impossível em casos normais devido aos conflitos entre as funções objetivas. Assim, Song *et al.* (2018) exploram a utilização de um conjunto de soluções ótimas chamado de Pareto, portanto, não apenas a convergência, mas também a diversidade do conjunto de soluções é explorada. Finalizando, um método de tomada de decisão baseado em *clustering* (agrupamento) foi introduzido para selecionar soluções finais das estruturas de proteínas geradas. Venske *et al*. (2016) também relata sucesso em explorar o uso de funções multiobjetivo para a resolução do problema. No caso, os autores construíram um algoritmo incorporando conceitos de evolução diferencial adaptativa baseado em decomposição, o qual,

segundo os próprios autores, tende a ser mais eficiente do que outras técnicas em problemas complexos, uma vez que os parâmetros são autoajustáveis, dispensando assim, a necessidade de um profissional da bioinformática para este trabalho.

No ano de 2020 ocorreu o CASP14, no qual o algoritmo AlphaFold da empresa DeepMind, ramificação da Google IA, surpreendeu a todos com uma acurácia de mais de 90% na previsão das estruturas 3D de determinadas proteínas especificadas na competição. Na competição anterior, em 2018 o grupo já havia conseguido superar seus concorrentes ao conseguir uma resposta de quase 60% de correspondência com o dado experimental. O AlphaFold utiliza CNNs para treinamento. Sequencias de aminoácidos idealizadas em duas matrizes, correspondendo à distribuição das distancias e aos ângulos de torção. Posteriormente, a resposta dessa parte baseada em aprendizado profundo é traduzida para o modelo 3D da proteína utilizando um método de gradiente iterativo descendente (Senior *et al.,* 2020).

3 METODOLOGIA

Este capítulo descreve o algoritmo de otimização Jaya desenvolvido para realizar a predição de estruturas de proteínas utilizando a modelagem computacional AB *Off-Lattice* e método *ab-initio*. A escolha de utilizar a modelagem AB *off-lattice*, baseia-se nas características desse modelo, tais como o fato deste não considerar apenas o efeito da interação da estrutura da proteína entre dois aminoácidos adjacentes, mas também os efeitos não locais entre aminoácidos não adjacentes (Stillinger *et al*., 1993), fazendo-o ser mais próximo das estruturas reais de proteínas quando comparado ao modelo HP *lattice*.

3.1 FUNÇÃO OBJETIVO

Simplificadamente, a função objetivo precisa imitar a função de energia livre da proteína, descobrindo assim as estruturas com energia livre mínima. Essa última combina entalpia e entropia, sendo uma quantidade estatística a qual depende de um conjunto de estruturas, assim, formular uma função objetivo para o problema de PSP é uma tarefa complexa. As funções objetivo podem ser classificadas em três tipos, as baseadas em física, baseadas em conhecimentos, ou as que são uma combinação dessas duas.

As funções objetivo baseadas na física são modelos matemáticos que descrevem as interações interatômicas da estrutura (Duan e Kollman, 1998). São descritas por termos relacionados ao comprimento e ângulo das ligações, iterações de van der Waals, interações eletrostáticas e efeitos hidrofóbicos. Algumas ainda são baseadas em características de solventes, preferências estruturais secundárias, preferências de ângulo de torção, potenciais de pares resíduo-resíduo, fração de empacotamento e são derivadas de estruturas de proteínas encontradas experimentalmente (Kaushik *et al*., 2018).

As funções baseadas em conhecimento, também chamadas de funções baseadas em estatística ou funções de energia empírica, são modelos estatísticos que provém diferentes parâmetros estatísticos derivados das frequências de interações em estruturas determinadas experimentalmente, vistas em proteínas nativas, como é o caso de características estruturais e padrões de interação,

derivados de conjuntos de dados de estruturas de proteínas determinadas experimentalmente (Kaushik *et al.,* 2018; Shen e Sali, 2006).

As funções integradas foram desenvolvidas para melhorar a precisão das previsões, acoplando as duas abordagens para diminuir a limitação que ambas têm separadas. Essa abordagem combina a extração dos recursos e padrões de estruturas proteicas encontrados experimentalmente em laboratórios e funções objetivas baseadas em físicas, a fim de identificar o correto dobramento da proteína (Singh *et al.*, 2016).

O modelo AB *Off-Lattice* utiliza de noções de física para ser correspondida, uma vez que com base no ponto de vista de que a estrutura nativa de uma proteína corresponde a estrutura entre as possíveis com o menor valor de energia livre. Esta energia consiste em duas partes, uma é a interação intermolecular entre os átomos de proteína, e o outro é a interação intermolecular entre as proteínas e o ambiente circundante moléculas de solvente (Anfinsen, 1973)

Neste modelo, os vinte tipos de são classificados entre resíduos hidrofóbicos e hidrofílicos. As partículas se conectam por ligações químicas e, portanto, formam uma corrente não direcional. A conformação de qualquer corrente com n partículas é especificada pelos n - 2 ângulos de curvatura [$\theta_1; \theta_2; \ldots; \theta_{n-2}$] como mostrado na Figura 2.11. É arbitrariamente definido que $\theta_i \in [-180^o; 180^o)$ para cada ângulo de dobra neste modelo. Notavelmente, $\theta_i \in [-180^o; 0^o)$ significa que existe uma tendência de rotação no sentido anti-horário na cadeia, e $\theta_i \in (0^o; 180^o)$ indica uma tendência de rotação no sentido horário.

A função de energia livre de uma sequência de aminoácidos é definida por

$$Energia = \sum_{i=1}^{n-2} \frac{1-\cos\theta_i}{4} + 4\sum_{i=1}^{n-2}\sum_{j=i+2}^{n}[r_{ij}^{-12} - C(\xi_i, \xi_j)r_{ij}^{-6}]. \qquad (3.1)$$

A propriedade da i-ésima partícula individual é refletida por ξ_i. Se o resíduo i for hidrofílico, então ξ_i é igual a um, caso contrário, ξ_i é igual a menos um. r_{ij} denota a distância entre as partículas i e j na cadeia.

$$r_{ij} = \sqrt{\left[1 + \sum_{k=i+1}^{j-1}\cos\left(\sum_{l=i+1}^{k}\theta_l\right)\right]^2 + \left[\sum_{k=i+1}^{j-1}\mathrm{sen}\left(\sum_{l=i+1}^{k}\theta_l\right)\right]^2} \qquad (3.2)$$

$C(\xi_i, \xi_j)$ representa a interação entre duas partículas e é dada pela Equação (2.5). Esse coeficiente se iguala a 1 para pares hidrofóbicos (AA), 0,5 para pares hidrofílicos (BB) e -0,5 para pares diferentes (BA ou AB). É baseado na suposição de que as correlações entre as partículas hidrofóbicas devem ser fortemente reforçadas, enquanto as partículas hidrofílicas são fracamente encorajadas, de outra forma, resulta em uma repulsão fraca.

3.2 PROTEÍNAS ESCOLHIDAS PARA O ESTUDO

A Tabela 3.1 apresenta as sequências de proteínas utilizadas nessa pesquisa. A primeira coluna é o índice que será utilizado para representar cada uma das dez proteínas nessa pesquisa. A segunda coluna representa o código da proteína no PDB. Já a terceira se refere ao comprimento em número de aminoácidos da proteína, enquanto a coluna de número quatro é o número de dimensões a serem otimizadas na proteína. Por fim, a última coluna representa a sequência de aminoácidos na forma AB *Off-Lattice* de acordo com sua hidro familiaridade, dessa forma os aminoácidos A, C, G, I, L, M, P e V são considerados hidrofóbicos (A) e R, N, D, E, Q, H, K, F, S, T, W e Y são representados pela característica de hidrofilia (B).

Tabela 3.1 Sequências das proteínas

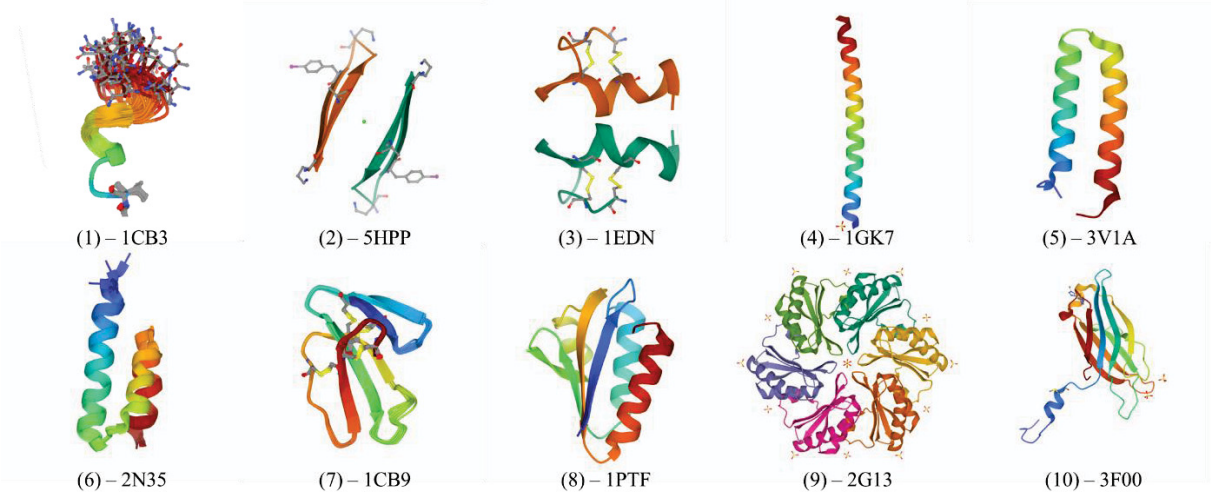
Nº	Código PDB	Tamanho	Dimensão	Sequência
1	1CB3	13	21	BABBBAABBAAAB
2	5HPP	16	27	ABAAAAABABBBBBAA
3	1EDN	21	37	ABABBAABBBAABBABABAAB
4	1GK7	39	73	ABBBBABABBABBBBABBABBABBABBBBBAAAABABBA
5	3V1A	48	91	ABABAAABBABBABBBABBABAAABABBABBABBBABBAABBBBBBBBB
6	2N35	52	107	ABABBBAABBABBBAABBABBAAABBBBBBBAABBABBABAABAABBBAABA
7	1CB9	60	115	ABABBAAAABBBBAAAABBAABBABAAAAABAAABBAAABAAABBBAAABBAAABBBBAB
8	1PTF	88	171	ABBBBBBAAABBAABABAABAAABBABBBBBBABABBBABBABABBAAAAABAAAABABBABABABAABB ABAAAAAABBABBBAAAB
9	2G13	98	191	AABABAAAAAAABBBAAAAAABAABAABBAABABAAABBBAAAABABAAABABBAAABAAABAAABAABB AABAAAAABAAABABBBABBAAABAABA
10	3F00	143	281	ABAAAAAAAAAAABBAABBAABABBBABBBBBBBBAAAAAABAABAAAABAAABBBABABABAAABBBBBB BBBABBBBABAABBBBBBBBBAABBBAAABBAAABBAAAAABBBBBBBBBBAAABBBAAABBABBABABBBBBBABBA

Fonte: a autora, 2021.

A Figura 3.1 mostra a representação de tais proteínas como são relatadas na literatura e de acordo com o PDB. Assim, ao invés de serem representadas por

cada aminoácido que as compõe, estas são representadas pelas α-hélices e β-folhas, assim como pelos átomos que compõe cada aminoácido, os quais podem estarem conectados por ligações entre cadeias que não são importantes para este projeto. Por isso nessas representações as proteínas aparentam ser mais complexas do que na representação *AB Off-Lattice*.

Figura 3.1 - Proteínas escolhidas



Fonte: PDB, adaptado, 2021.

3.3 MÉTRICA DE AVALIAÇÃO DA ESTRUTURA - RMSD

Uma métrica relevante quando se trata de PSP via métodos computacionais se refere à comparação da estrutura predita em relação a uma estrutura nativa original, com a finalidade de avaliar quão similar é a adequação predita. Uma métrica frequentemente utilizada para esses casos é o desvio médio quadrático (do inglês *Root Mean Square Deviation,* RMSD), a qual avalia o grau de semelhança entre as estruturas em *Angstroms* (Å) ou, como é mais comumente utilizado, em nanômetros (nm) (Xu *et al*., 2006). O RMSD pode ser encontrado pela equação a seguir

$$RMSD(a,b) = \sqrt{\frac{\sum_{i=1}^{n} |r_{ai} - r_{bi}|^2}{n}} \tag{4.1}$$

na qual, r_{ai} e r_{bi} são as posições do átomo *i* nas estruturas a serem comparadas *a* e *b*, respectivamente, e *n* é o número de átomos envolvidos. Consequentemente, estruturas idênticas possuem seu valor de RMSD nulo. Quanto maior o valor de RMSD encontrado, mais divergentes são as estruturas (Moss *et al*., 2005). Assim, um RMSD entre 3 e 5 Å é considerado satisfatório e acima disso não é informativo a respeito da proteína.

4 APRESENTAÇÃO DOS RESULTADOS

As simulações foram realizadas com um tamanho de população de 50 em 1000 gerações. A ferramenta utilizada para rodar o algoritmo foi o ambiente computacional MatLab e o hardware sendo um processador AMD Ryzen 5 3600X 6-Core unidade de central de processamento (do inglês *central processing unit*, CPU) @ 3.79GHz e computador de 16GB de memória de acesso volátil (do inglês *random access memory*, RAM). Para cada proteína o algoritmo foi rodado 50 vezes. Além disso a melhor solução foi salva e usada para criar o modelo visual da proteína.

Cabe mencionar que as imagens das conformações são representações 3D das proteínas, por isso elas podem não se parecer com as proteínas originais, conforme apresentado na Figura 3.1, uma vez que as primeiras são representadas apenas pelos aminoácidos que as compõem, representados por esferas, além de que foi escolhido o melhor ângulo para se ver todos os aminoácidos, assim a rotação da proteína pode estar diferente do que da imagem original.

A Tabela 4.1 expõe os resultados obtidos para cada um dos algoritmos e compara-os com outros algoritmos já consolidados nessa área, o DE, a otimização de colônia de abelhas artificial (do inglês *Artificial Bee Colony*, ABC) e a otimização baseada no lobo cinza (do inglês *Gray Wolf Optimization*, GWO). Em sequência, os resultados para cada proteína são mais bem explicados.

Tabela 4.1 - Resultados das otimizações

Proteína	Métrica	Jaya	LJA	EO-Jaya	ABC	DE	GWO
(1) -1CB3	Mínimo	-2.5301	-3,4911	-3,3357	-4,4743	-4,8142	**-6,7320**
	Máximo	1.7482	-0.7516	0,8861	**-2,3438**	-0,3889	-0,9739
	Média	-0,1222	-1,8596	-1,7271	**-3,4785**	-2,0748	-3,3408
	Mediana	-0,0079	-1,8622	-1,8290	-3,4463	-2,5449	**-3,6848**
	Desvio padrão	0,6160	0,5440	0,8263	**0,5258**	1,5504	1,5003
	RMSD	4,9961	3,7933	3,8653	3,6817	3,6782	**3,0321**
(2) - 5HPP	Mínimo	-2,2484	-6,0471	-7,2662	-11,6236	-8,2748	**-14,8640**
	Máximo	1,8956	-3,0464	-0,7683	**-7,8910**	-2,1142	-5,4628
	Média	-0,0041	-4,6053	-3,6726	**-9,3517**	-3,9839	-8,8896
	Mediana	-0,0665	-4,5176	-3,7419	**-9,2190**	-3,7067	-8,3614
	Desvio padrão	1,0479	**0,6863**	1,1259	0,8394	1,4495	2,1527
	RMSD	4,8498	4,5495	3,8987	3,7985	3,8658	**3,1548**
(3) – 1EDN	Mínimo	0,1555	-0,9968	-2,8687	-9,3800	-5,2986	**-9,6331**
	Máximo	5,5805	1,2928	4,4175	**-5,1053**	-1,5434	-2,0909

	Média	3,6624	0,3715	-1,6501	**-6,6890**	-3,6764	-4,8805
	Mediana	3,7833	0,4932	-2,0040	**-6,6083**	-3,6135	-4,7514
	Desvio padrão	1,1298	**0,5749**	1,5842	0,9052	0,8573	1,5694
	RMSD	4,9998	4,8594	3,9035	3,5216	3,4267	**3,1235**
(4) - 1GK7	Mínimo	-4,3451	1,4181	-3,8236	**-10,7751**	3,2778	-8,1839
	Máximo	28,0281	15,8906	21,9626	**-5,8189**	11,2421	-3,0367
	Média	4,6214	7,7416	9,5615	**-7,9512**	6,2479	-5,2136
	Mediana	1,2461	7,7381	15,1808	**-8,0085**	6,0922	-4,9258
	Desvio padrão	7,8569	2,3379	8,8761	**0,9407**	2,1560	1,2061
	RMSD	5,5215	6,5187	5,8642	**4,6687**	7,6984	4,9587
(5) - 3V1A	Mínimo	-1,8688	8,3591	-4,7230	**-13,7700**	10.4566	-9,5950
	Máximo	377,8650	65,6789	23,2560	**-7,0682**	717,1670	-3,7250
	Média	15,5976	23,6346	11,6691	**-9,4971**	252,2555	-6,3764
	Mediana	2,6378	21,8151	18,7722	**-9,1643**	211,6021	-6,3765
	Desvio padrão	55,5438	10,4862	10,4565	1,5311	186,5387	**1,4176**
	RMSD	6,5647	10,2488	5,6987	**4,5218**	11,9854	5,0654
(6) - 2N35	Mínimo	-1,7176	15,0968	-3,1691	**-11,1002**	28.4101	-11,0898
	Máximo	274,7610	134,4600	27,9064	**-4,4199**	3.2439e+03	-1,9373
	Média	19,5576	50,3652	14,2331	**-7,9084**	7.0788e+02	-5,4869
	Mediana	4,3130	42,2600	22,4074	**-7,9421**	5.7491e+02	-5,1914
	Desvio padrão	41,5134	29,8567	12,0098	**1,6396**	6.2682e+02	2,2381
	RMSD	7,5124	12,3597	6,9129	**5,5874**	15,2947	5,5998
(7) - 1CB9	Mínimo	-4,1847	25,6327	**-32,5978**	-31,7102	1.2621e+02	-24,8148
	Máximo	1,0854e+03	1,0500e+03	20,3802	**-16,1547**	1.0936e+04	-3,9586
	Média	52,7178	207,7000	10,0499	**-21,1656**	3.3532e+03	-13,0408
	Mediana	5,6672	146,3895	2,3587	**-20,6889**	2.6793e+03	-12,0319
	Desvio padrão	163,4296	183,6267	14,7199	**3,0874**	2.4586e+03	4,6415
	RMSD	9,5487	15,6478	**5,7854**	5,8497	146,5878	6,5978
(8) - 1PTF	Mínimo	-1,8455	1,7955e+03	**-29,16547**	-24,8497	1,7120e+04	-29,0758
	Máximo	7,8772e+04	6,4061e+04	35,4552	**-11,3880**	2,0873e+05	-7,5016
	Média	3,3261e+03	1,8929e+04	15,5363	**-17,9879**	9,0570e+04	-14,6338
	Mediana	8,1921	1,7207e+04	5,4662	**-17,5595**	8,2593e+04	-14,1263
	Desvio padrão	1,3554e+04	1,1943e+04	14,4789	**3,0361**	4,7587e+04	4,0791
	RMSD	15,4875	1248,1547	**6,5874**	7,0548	15247,2548	6,5987
(9) - 2G13	Mínimo	-2,5364	8,6702e+03	**-45,0790**	-44,6487	3,7934e+04	-32,2353
	Máximo	5,1709e+04	1,2685e+05	379,1108	**-25,3592**	6,3039e+05	-13,0546
	Média	2,0495e+03	5,1005e+04	24,0041	**-32,4497**	2,1281e+05	-21,3403
	Mediana	5,9583	4,6264e+04	5,2995	**-32,2803**	1,7669e+05	-21,5634
	Desvio padrão	8,4986e+03	2,8504e+04	57,3192	4,5076	1,3236e+05	**4,0203**
	RMSD	31,2548	2648,278	**10,2658**	10, 5487	74872,6589	12,9659
(10) - 3F00	Mínimo	4,5702	9.8963e+04	-28,2219	**-29,6048**	2,8770e+05	-28,9738
	Máximo	1,8756e+06	2.7958e+06	4,9672e+05	19,5107	6,2608e+06	**-11,9113**
	Média	1,0219e+05	1.0665e+06	1,6167e+04	-9,0343	2,6188e+06	**-19,9311**
	Mediana	16,3446	1.0506e+06	11,5678	-10,2247	2,5456e+06	**-19,5445**

Desvio padrão	3,4577e+05	5.1825e+05	8,2071e+04	9,1673	1,3164e+06	**4,1051**
RMSD	45,2187	8547,2484	26,6187	**25,8794**	95874,2648	26,5875

4.1 PROTEÍNA (1) – 1CB3

A proteína 1CB3 é a menor das proteínas trabalhadas e, portanto, a mais simples. Os algoritmos tiveram resultados parecidos, com todos os valores de RMSD dentro do limite considerado aceitável. O GWO obteve o melhor resultado entre todos as otimizações, contudo o algoritmo com menor desvio padrão, logo com os resultados mais concisos, foi o ABC, diferente do algoritmo DE, o qual teve o maior desvio padrão.

Entre os algoritmos baseados na otimização Jaya, o original teve o pior desempenho tanto em questão de valor RMSD quanto de minimização, sendo o com o pior valor mínimo entre todos os algoritmos. LJA teve o segundo melhor desvio padrão, mostrando ser mais preciso entre as otimizações baseadas em Jaya, além de ter o melhor valor mínimo entre os três também. EO-Jaya teve um desempenho mediano, com uma minimização coerente, porém com um desvio padrão maior.

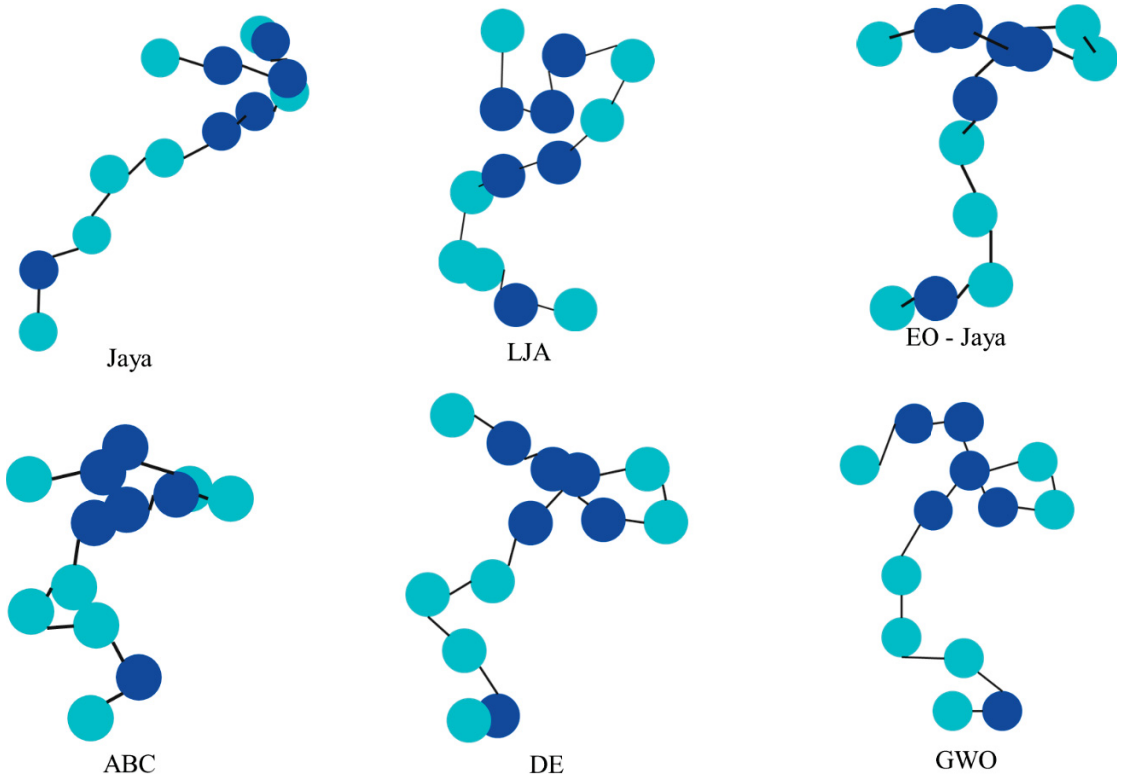
A Figura 4.1 representa as conformações da proteína com a menor energia livre encontrada para cada otimização. É claramente notável a semelhança entre as conformações, especialmente a apresentação do núcleo apolar na parte superior de cada proteína (aminoácidos indicados pela cor azul escura) e os aminoácidos hidrofílicos mais presentes no exterior da proteína como forma de “proteger” os aminoácidos hidrofóbicos.

A Figura 4.2 apresenta a convergência dos algoritmos de acordo com a média de cada iteração e é possível notar que todas as otimizações conseguiram, em média, convergir para o resultado final antes da 100ª iteração. ABC teve o valor inicial mais alto, demorando um pouco mais para convergir do que os demais, enquanto o DE foi o mais rápido para conversão com o menor valor inicial também. Os algoritmos baseados em Jaya obtiveram curvas semelhantes entre si.

A Figura 4.3 mostra o desempenho dos algoritmos com seus melhores resultados. Nota-se que o ABC foi o que convergiu mais rápido, seguido pelo Eo-Jaya. É possível verificar também como todos os algoritmos se comportam de forma semelhante, com vários “degraus”, nos quais os algoritmos permanecem em um

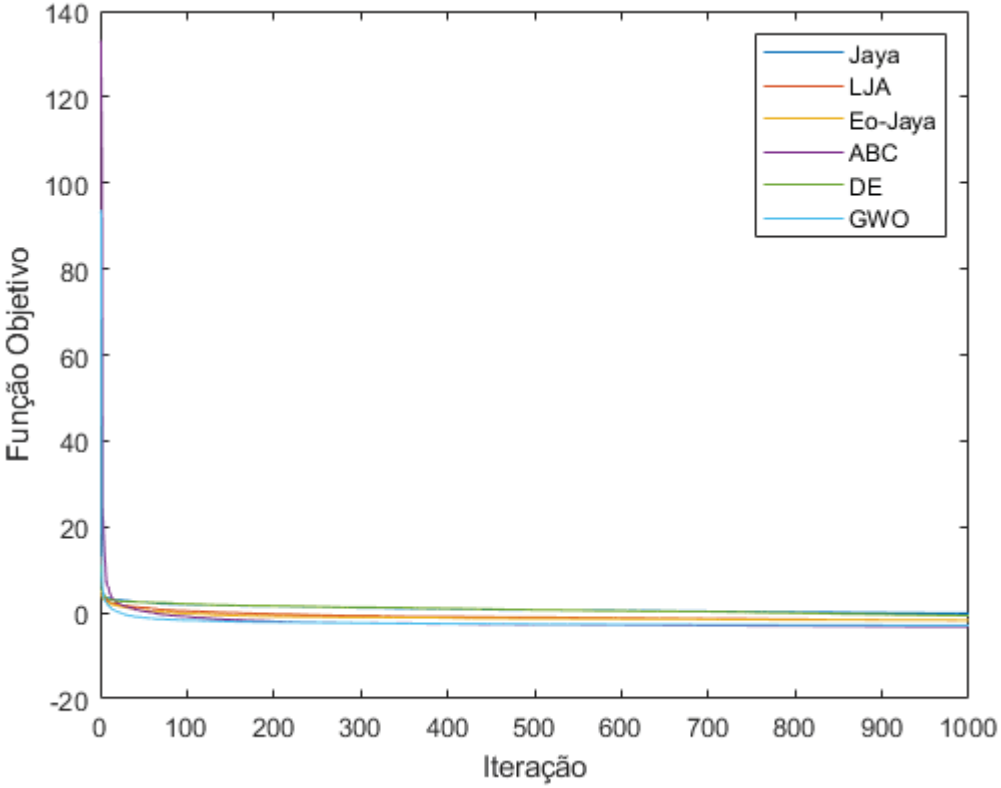
mesmo ponto por um tempo. Assim, observa-se que LJA e EO-Jaya possem maior capacidade de sair de um ótimo local do que o algoritmo de Jaya original.

Figura 4.1 - Melhores conformações para a proteína 1CB3



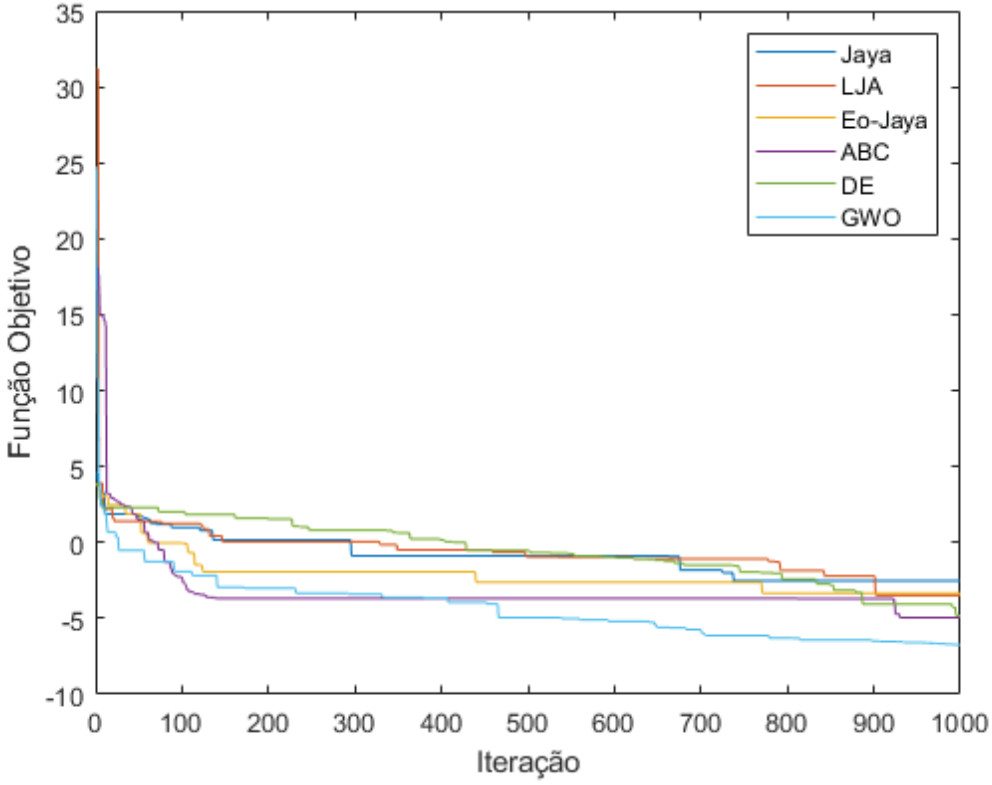
Fonte: a autora, 2021.

Figura 4.2 - Convergência da média de cada iteração para a proteína 1CB3



Fonte: a autora, 2021.

Figura 4.3 - Convergência do melhor resultado de cada algoritmo para a proteína 1CB3



Fonte: a autora, 2021.

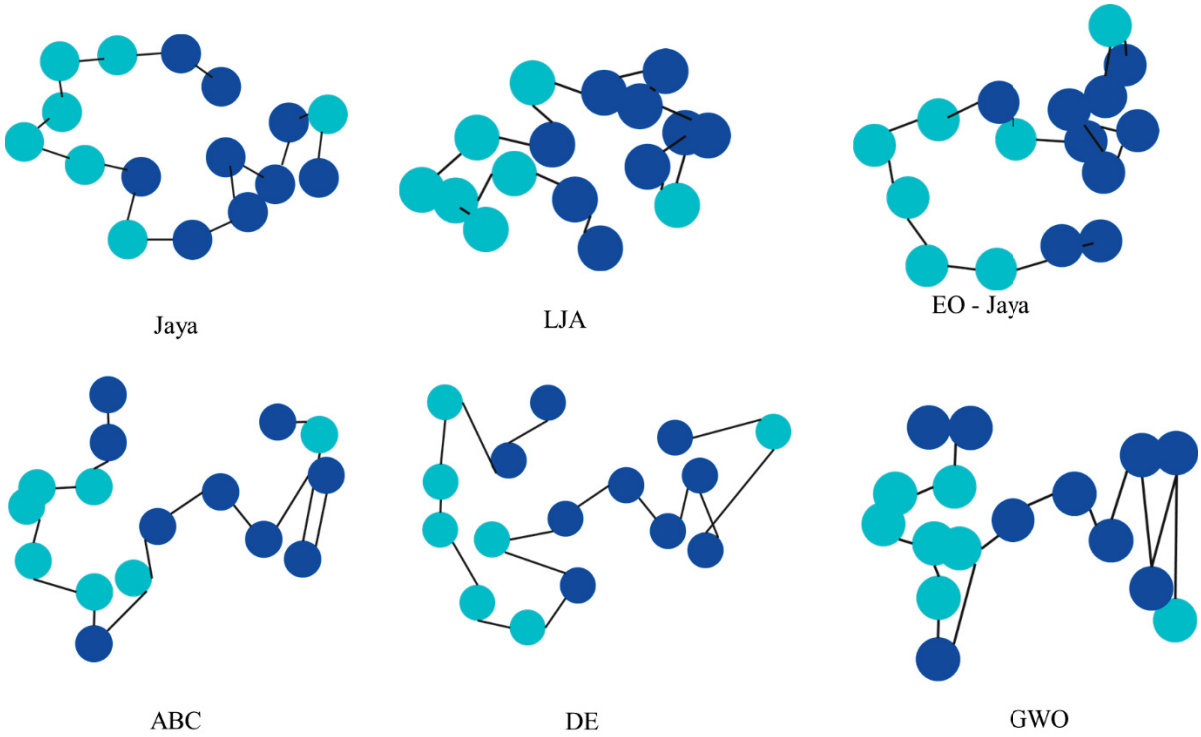
4.2 PROTEÍNA (2) – 5HPP

Assim como para a primeira proteína, os resultados das otimizações foram parecidos, como é visto na Tabela 4.1. O melhor desempenho foi novamente do algoritmo GWO, possuindo o menor valor mínimo encontrado e por conseguinte o menor valor RMSD. Os algoritmos ABC e DE tiveram também bons resultados, sendo os que tiveram o segundo e terceiro menor valor de minimização, respectivamente. O melhor valor de média de resultados foi do algoritmo ABC.

Dentre os algoritmos baseados na otimização Jaya, o original novamente teve o pior desempenho. EO-Jaya foi o melhor entre os três em relação ao valor mínimo, sendo que o LJA teve valores muito próximos e foi o com menor média e mediana. O LJA teve o menor desvio padrão entre todos os algoritmos, sendo assim o mais conciso nesse caso.

A Figura 4.4 representa as conformações da proteína 5HPP com a menor energia livre encontrada para cada otimização. Como todos os valor de RMSD foram próximos, é possível notar a semelhança entre as seis conformações, além do claro núcleo apolar que se forma na região mais à direita da proteína.

Figura 4.4 - Melhores conformações para a proteína 5HPP

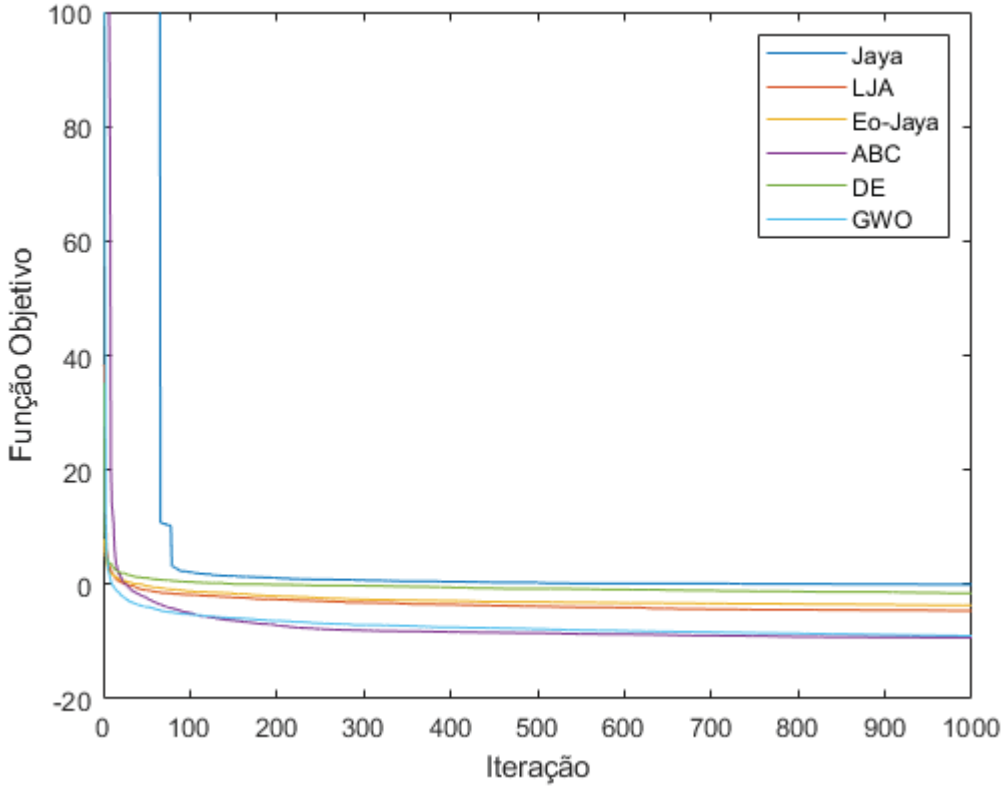


Fonte: a autora, 2021.

A Figura 4.5 apresenta a convergência dos algoritmos de acordo com a média de cada iteração e é possível notar que todas as otimizações conseguem, em média, se manter estáveis a partir da 200ª iteração. O algoritmo original de Jaya teve o valor inicial mais alto, em torno de 140000, muito acima dos demais e por isso sua curva parece cortada inicialmente e é o algoritmo que demora mais para se estabilizar. Os algoritmos GWO e ABC possuem curvas parecidas, assim como as duas variantes de Jaya.

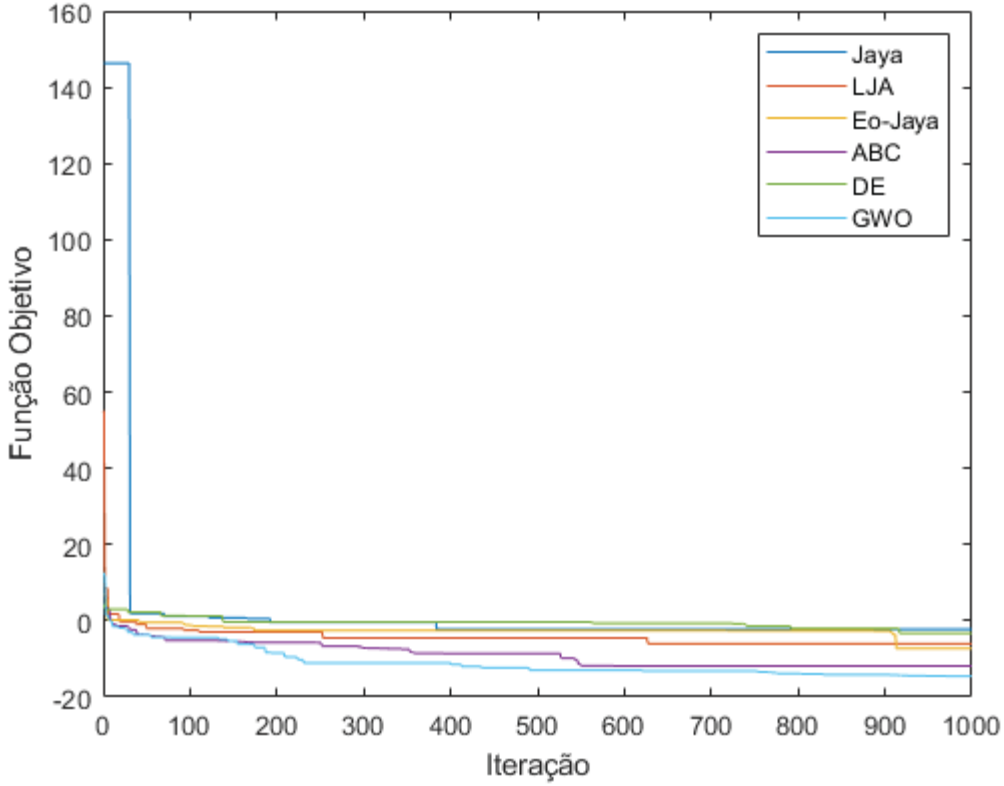
A Figura 4.6 mostra o desempenho dos algoritmos com seus melhores resultados. Nota-se que o Jaya foi o que convergiu mais rápido, mesmo que tenha ficado nas primeiras iterações preso em um mínimo local por mais tempo que os demais. GWo também teve uma convergência rápida. Observa-se também que LJA e EO-Jaya possem maior capacidade de sair de um ótimo local do que o algoritmo de Jaya original. O algoritmo DE foi o que teve menor aprimoração dos resultados por geração.

Figura 4.5 - Convergência da média de cada iteração para a proteína 5HPP



Fonte: a autora, 2021.

Figura 4.6 - Convergência do melhor resultado de cada algoritmo para a proteína 5HPP



Fonte: a autora, 2021.

4.3 PROTEÍNA (3) – 1EDN

A partir da terceira proteína começa-se a notar que os resultados apresentados na Tabela 4.1 começam a se divergir mais para cada algoritmo. GWO teve o melhor valor de energia mínima e o melhor RMSD, logo seguido pelo ABC, o qual teve os menores valores de média e mediana entre as otimizações. Assim, esses algoritmos se mostraram superiores que os demais, exceto para a métrica de desvio padrão, a qual teve seu melhor valor dado pelo algoritmo baseado em Jaya com voos de Lévy.

Dentre os algoritmos baseados em Jaya, o original novamente teve o pior resultado em todos os quesitos exceto desvio padrão. EO-Jaya, entre os três, teve o melhor valor de mínimo, média e mediana, mesmo assim, seu desvio padrão mostrou que este era a otimização com resultados menos concisos. LJA teve o melhor valor máximo entre os três e, como já dito, o melhor desvio padrão entre todos, mostrando-se ser coerente e preciso.

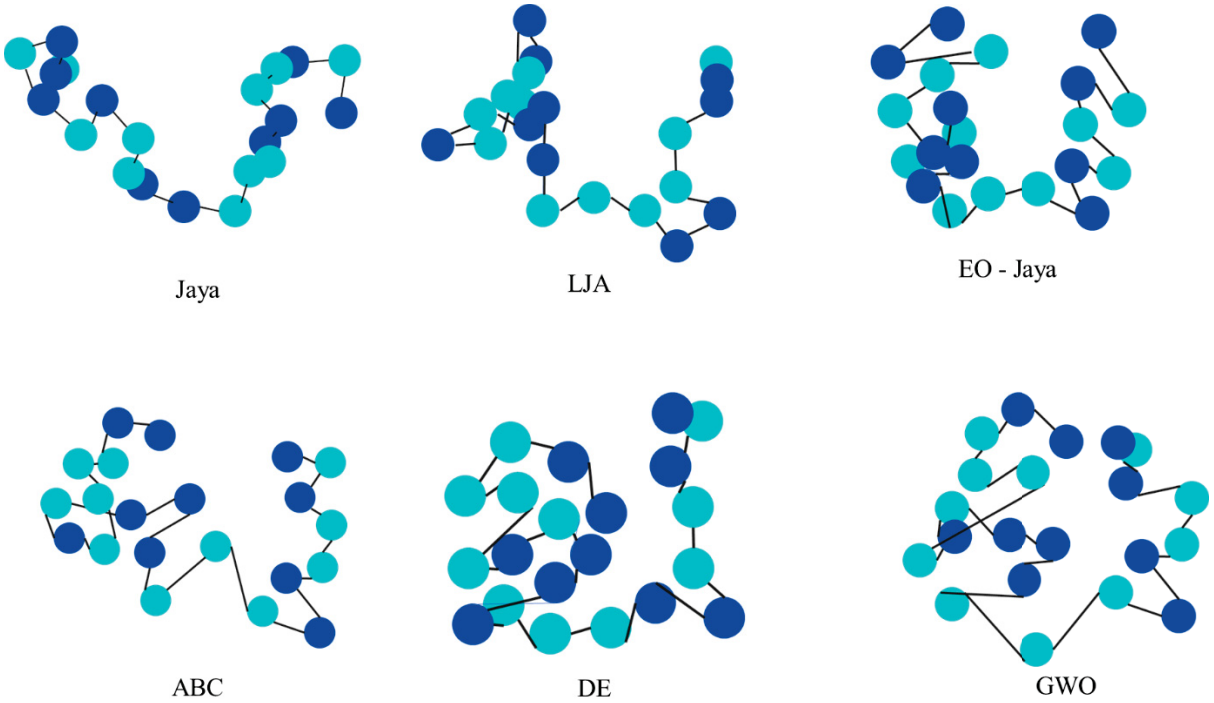
A Figura 4.7 representa as conformações da terceira proteína com a menor energia livre encontrada para cada otimização. Nota-se que as proteínas baseadas

em Jaya, especialmente a original, tiveram dificuldades em representar a ideia de núcleos hidrofóbicos, nas demais é fácil notar que os aminoácidos azuis escuros estão localizados principalmente na região central da proteína, sendo protegidos pelos aminoácidos hidrofílicos. Também é observável que os aminoácidos tendem a se colocar em “zigue-zague”, mostrando assim a natureza das α-hélices dessa proteína.

A Figura 4.8 apresenta a convergência dos algoritmos de acordo com a média de cada iteração e é possível notar que todas as otimizações normalmente convergem entre a 200ª e 300ª iteração. Jaya teve o valor inicial mais alto, demorando um pouco mais para convergir do que os demais, enquanto o GWO foi o mais rápido para conversão. O LJA e o EO-Jaya tiveram curvas médias muito parecidas.

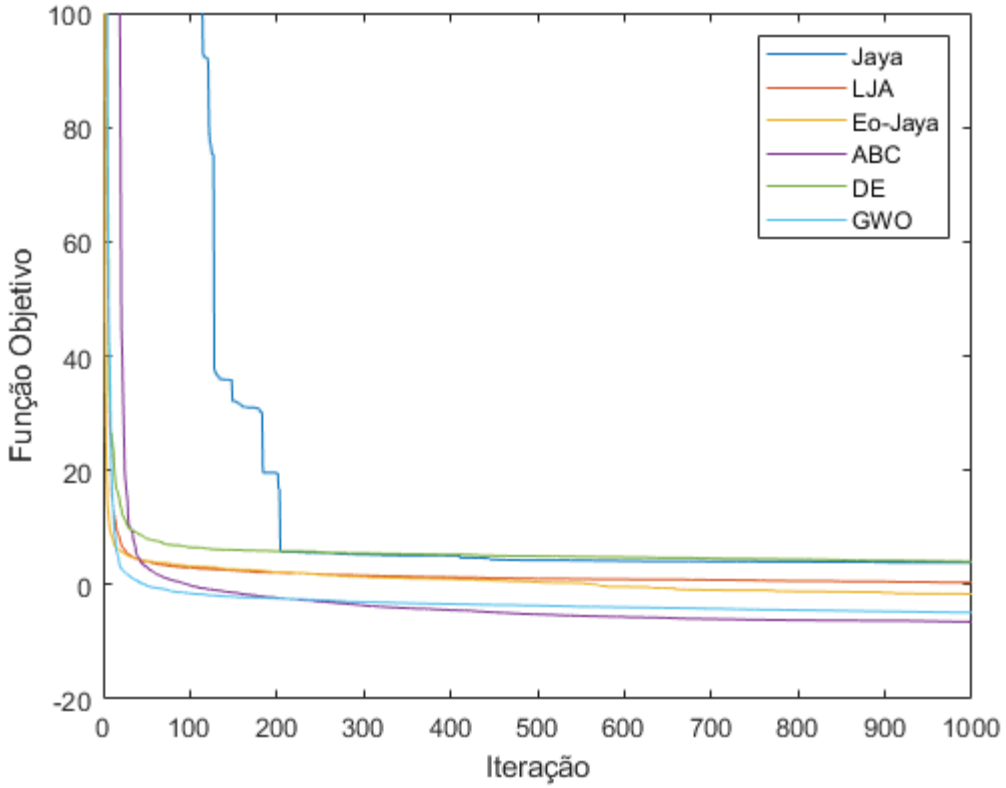
A Figura 4.9 mostra o desempenho dos algoritmos com seus melhores resultados. Nota-se que LJA e EO-Jaya foram os mais rápidos na conversão. Jaya foi o que teve maiores problemas para realizar a curva mais acentuada para um mínimo melhor.

Figura 4.7 - Melhores conformações para a proteína 1EDN



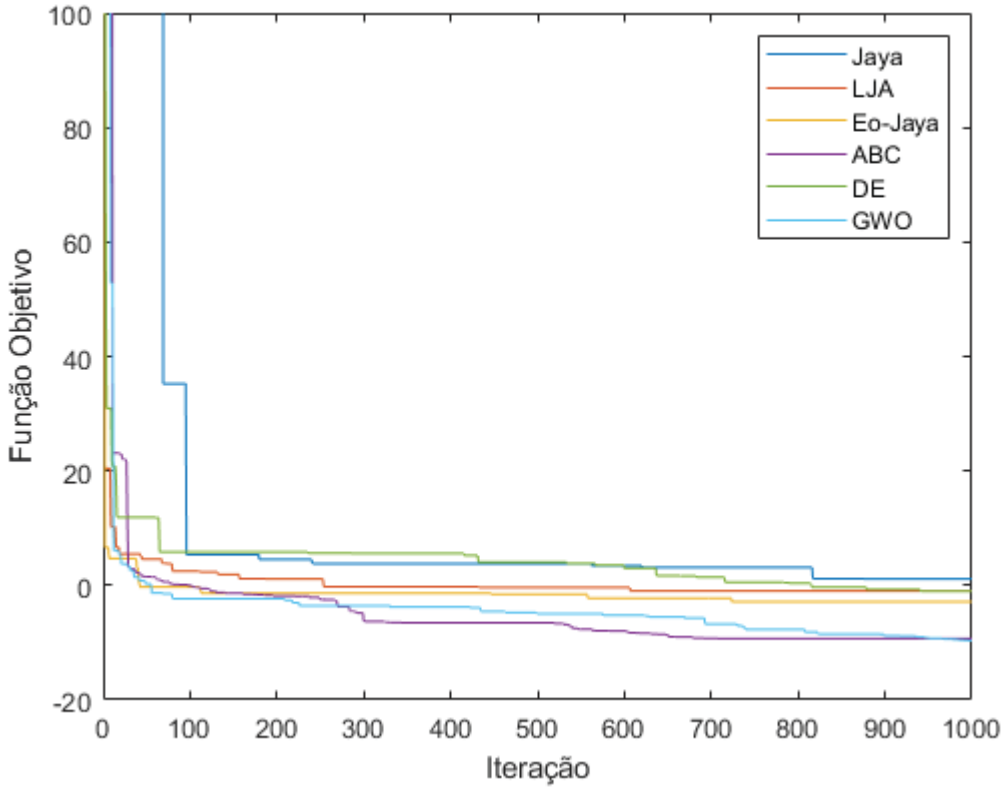
Fonte: a autora, 2021.

Figura 4.8 - Convergência da média de cada iteração para a proteína 1EDN



Fonte: a autora, 2021.

Figura 4.9 - Convergência do melhor resultado de cada algoritmo para a proteína 1EDN



Fonte: a autora, 2021.

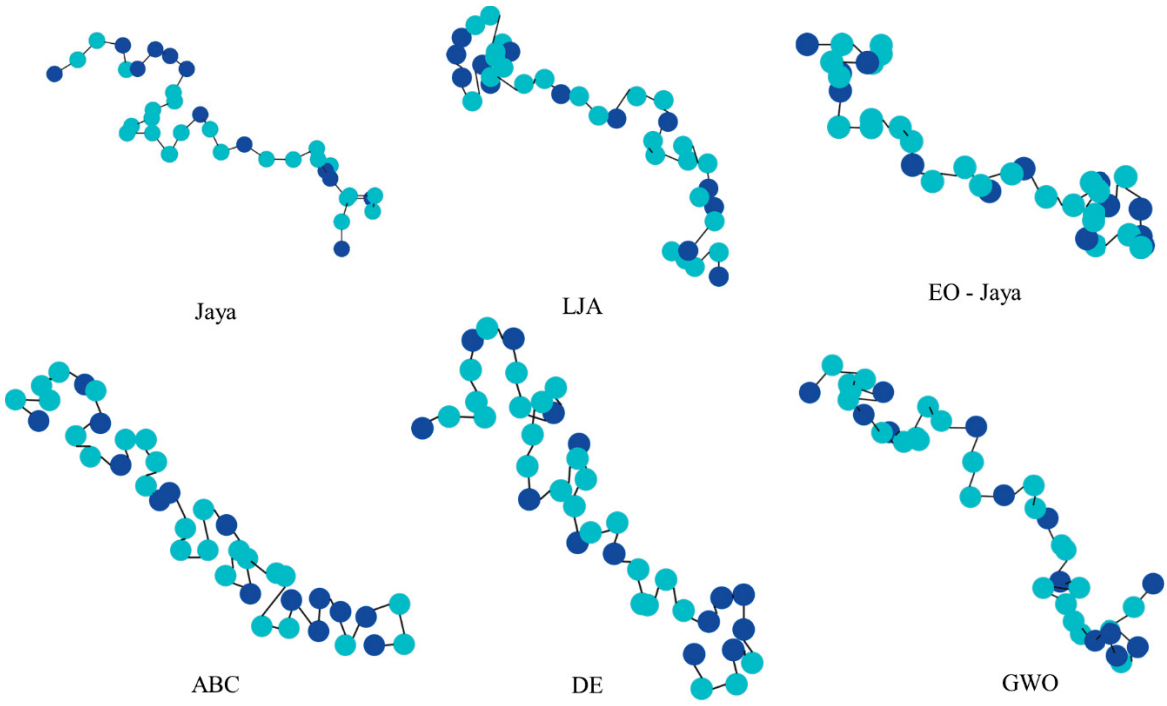
4.4 PROTEÍNA (4) – 1GK7

Em relação à proteína 1GK7, a otimização ABC obteve os melhores resultados em todas as métricas de acordo com a Tabela 4.1, seguido pelo GWO, sendo esses dois algoritmos os únicos que obtiveram valores de RMSD dentro do padrão satisfatório. Os demais algoritmos obtiveram valores de RMSD muito próximos do padrão, com exceção do DE com um RMSD acima de 7Å.

Dentre os algoritmos baseados em Jaya, diferente das proteínas anteriores, o pior resultado mínimo foi do LJA, contudo este teve o melhor desvio padrão entre os três. EO-Jaya obteve os maiores valores de média e mediana, enquanto o Jaya original se mostrou melhor que os outros nos nesses mesmos quesitos, além de ter o menor valor mínimo e por consequência o melhor RMSD.

A Figura 4.10 representa as conformações da proteína 1GK7 com a menor energia livre encontrada para cada otimização. É notável que nas melhores conformações há uma clara espiral formada pelos aminoácidos, refletindo o formato da α-hélice que constitui essa proteína. Também é notável a criação de um núcleo hidrofóbico em todas as proteínas.

Figura 4.10 - Melhores conformações para a proteína 1GK7

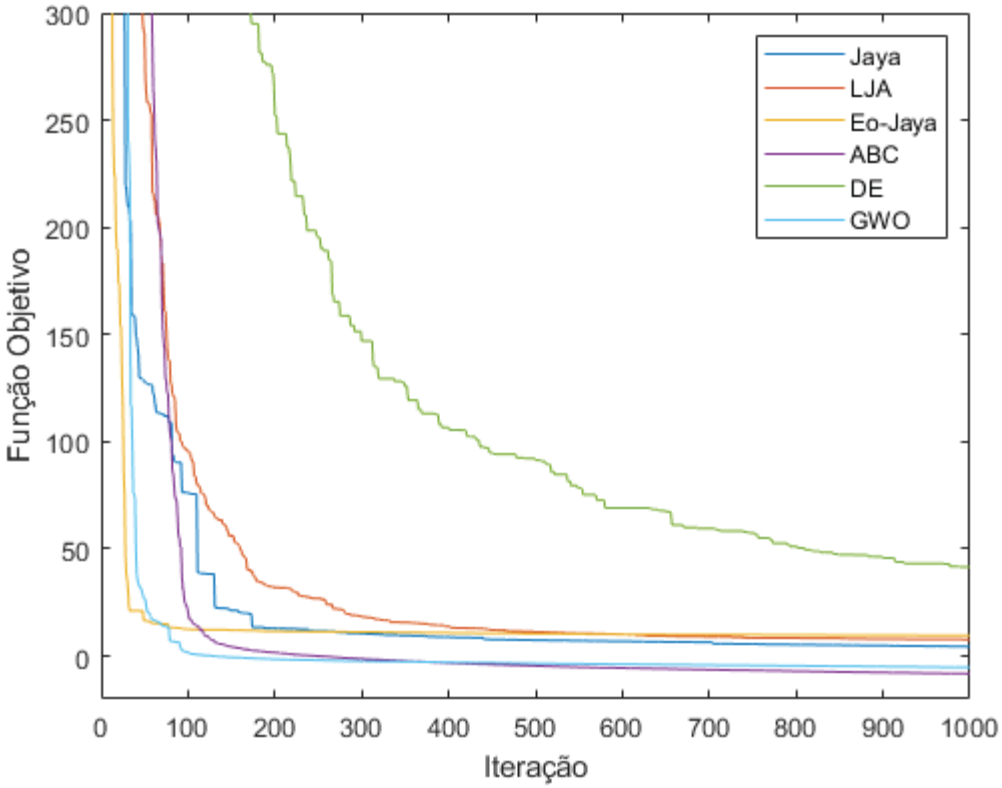


Fonte: a autora, 2021.

A Figura 4.11 apresenta a convergência dos algoritmos de acordo com a média de cada iteração e é possível notar que todas as otimizações normalmente convergem entre a 300ª e 400ª iteração, com exceção da DE, que parece que ainda não se estabiliza pro completo até a última iteração. Os três algoritmos baseados em Jaya obtiveram curvas médias bastante semelhantes, mas nota-se a superioridade do EO-Jaya em relação as outras, convergindo mais rápido.

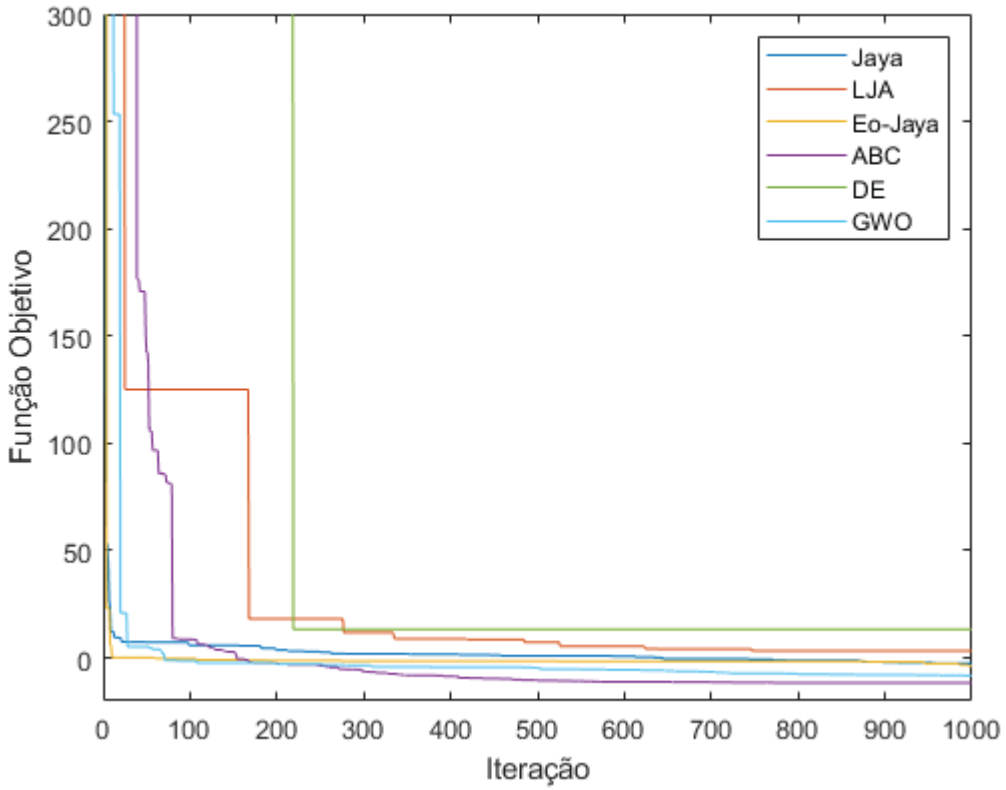
A Figura 4.12 mostra o desempenho dos algoritmos com seus melhores resultados. Nas curvas mais discrepantes são de DE e LJA, as quais demoraram mais para se estabilizar e tiveram valores iniciais maiores (que não é possível ver no gráfico porque esses valores são na faixa de 40000). EO-Jaya foi o mais rápido a se estabilizar, seguido pelo Jaya, mostrando a natureza mais ágil das otimizações baseadas em Jaya.

Figura 4.11 - Convergência da média de cada iteração para a proteína 1GK7



Fonte: a autora, 2021.

Figura 4.12 - Convergência do melhor resultado de cada algoritmo para a proteína 1GK7



Fonte: a autora, 2021.

4.5 PROTEÍNA (5) – 3V1A

De acordo com a Tabela 4.1, o algoritmo ABC foi o melhor em todas as métricas, exceto desvio padrão, quesito no qual a otimização GWO se mostrou superior. DE foi o pior algoritmo em todas as avaliações, tendo valores muito altos em relação às outras otimizações. Apenas ABC obteve um valor de RMSD dentro do limite de 5Å, mesmo que GWO tenha chegado muito perto desse valor também.

Sobre os algoritmos baseados em Jaya, o LJA teve um desempenho comparável ao DE, especialmente se tratando do valor mínimo encontrado, sendo assim um resultado ruim. EO-Jaya teve o melhor rendimento em todos as métricas, com exceção da mediana, quesito no qual o Jaya original teve o melhor resultado. Assim, nota-se que com proteínas acima de 30 aminoácidos a variante com voo de Lévy acaba se tornando menos propícia do que o Jaya original.

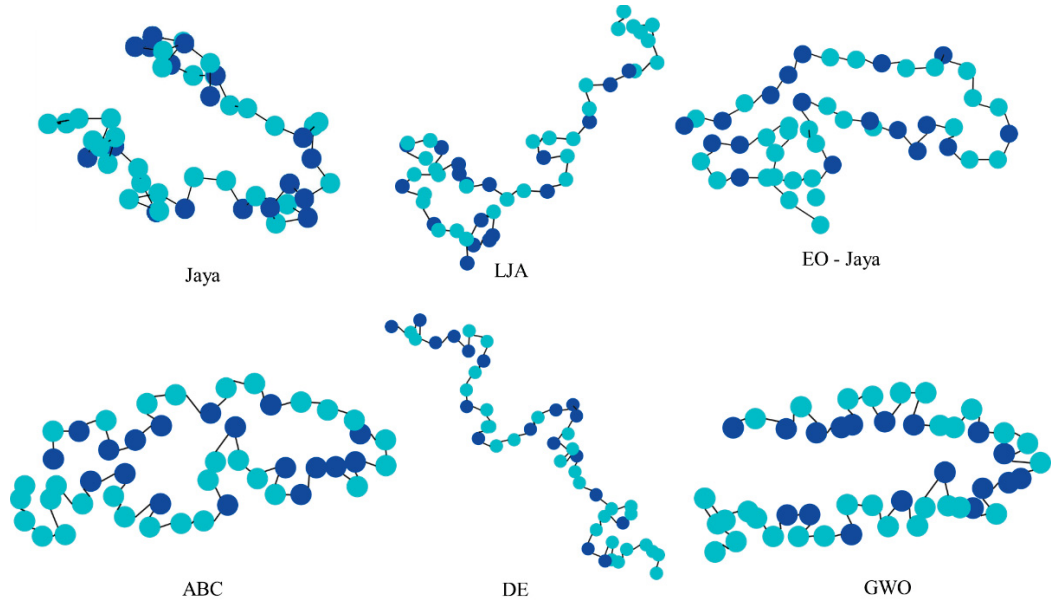
A Figura 4.13 representa as conformações da quinta proteína, 3V1A com a menor energia livre encontrada para cada otimização. Nas melhores conformações

(Jaya, EO-Jaya, ABC e GWO) é claramente notável a dobra que a proteína faz em forma de duas α-hélices, assim como a original tem de acordo com a Figura 3.1, o que não ocorre nas proteínas que tiveram valores RMSD maiores que 10 Å. É também possível observar a formação do núcleo hidrofóbico, especialmente nas proteínas conformadas pelos algoritmos ABC e GWO.

A Figura 4.14Figura 4.11 apresenta a convergência dos algoritmos de acordo com a média de cada iteração e é possível notar que as otimizações normalmente convergem até 400ª iteração, com exceção da DE, que parece que ainda não se estabiliza pro completo até a última iteração e de LJA que demora além da 600ª iteração para se estabilizar. EO-Jaya obteve uma curva muito parecida com a do melhor, se mostrando tão coerente como a GWO. Jaya teve uma curva parecida com a de ABC, porém mais lenta para convergir.

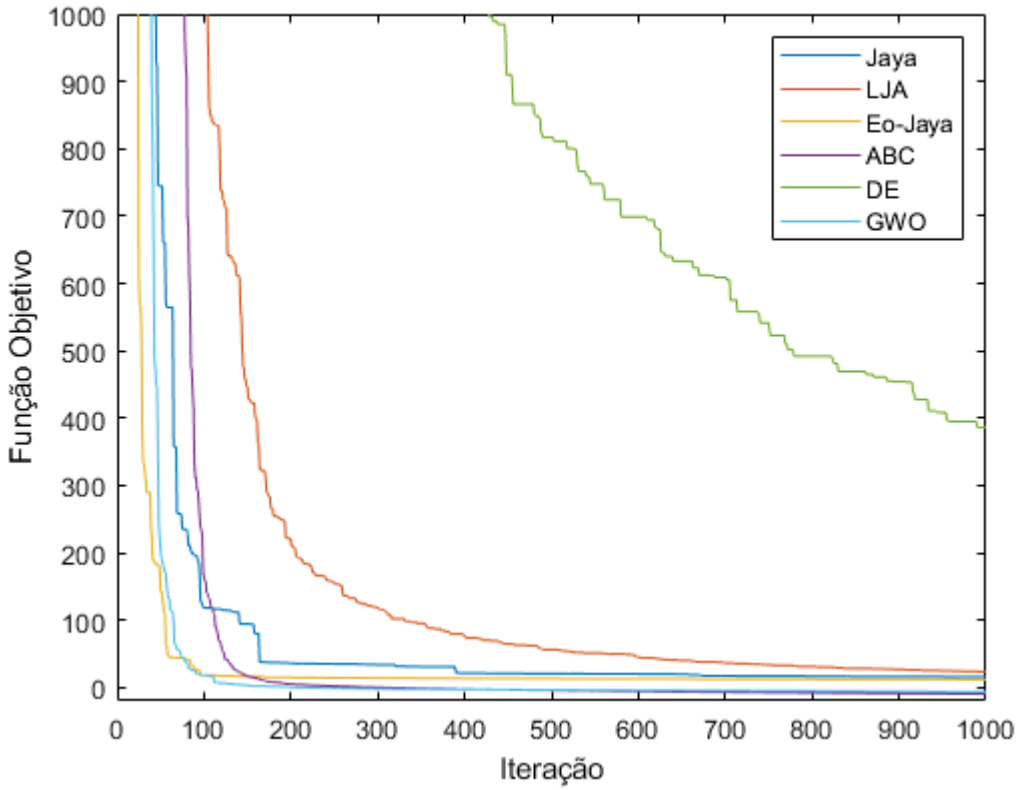
A Figura 4.15 mostra o desempenho dos algoritmos com seus melhores resultados. EO-Jaya e GWO novamente se mostram equivalentes, enquanto DE converge-se em uma curva bastante brusca, ou seja, de uma geração para outra, essa otimização encontra um valor muito menor e se mantem nele até o final das iterações, dando a ideia de ser mais factível a ficar presa em mínimos locais. LJA foi a que mais demorou a se estabilizar, mesmo assim teve um proveito melhor do que DE. Jaya e ABC também obtiveram curvas parecidas, sendo que a segunda convergiu antes da primeira. Nota-se então que EO-Jaya foi a melhor das três otimizações baseadas em Jaya, sendo equiparada pelas duas otimizações com melhores desempenhos (GWO e ABC).

Figura 4.13 - Melhores conformações para a proteína 3V1A



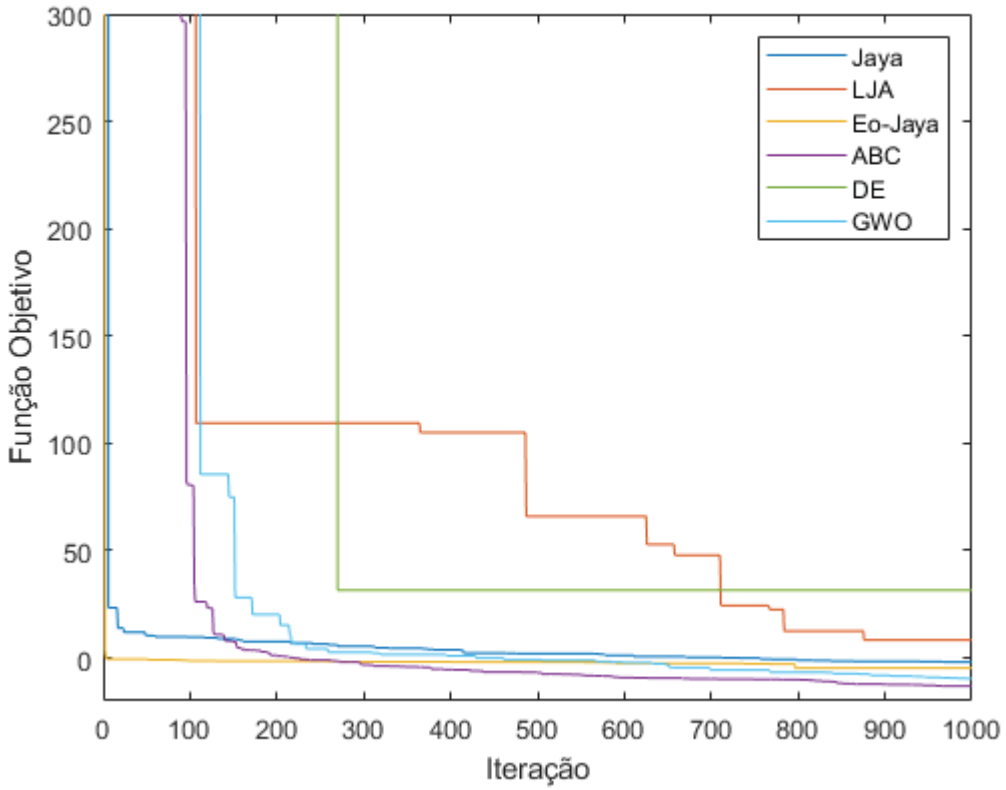
Fonte: a autora, 2021.

Figura 4.14 - Convergência da média de cada iteração para a proteína 3V1A



Fonte: a autora, 2021.

Figura 4.15 - Convergência do melhor resultado de cada algoritmo para a proteína 3V1A



Fonte: a autora, 2021.

4.6 PROTEÍNA (6) – 2N35

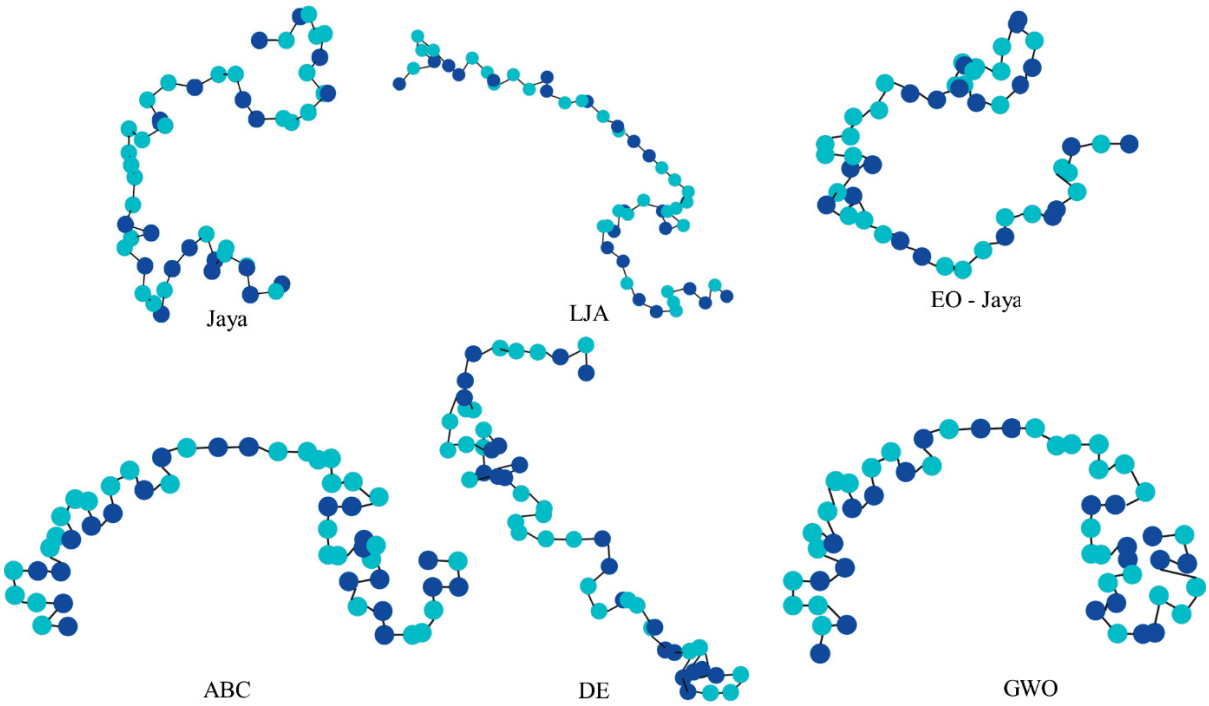
Assim como é visto na Tabela 4.1, o algoritmo ABC foi o melhor em todas as métricas, seguido pelo GWO que teve resultados bastante semelhantes. DE foi o pior algoritmo em todas as avaliações, tendo valores muito altos em relação às outras otimizações. Nenhum dos algoritmos teve valor RMSD dentro do padrão, mostrando que quanto maior a proteína maior a dificuldade de se encontrar a conformação certa.

Entre os algoritmos baseados em Jaya, EO-Jaya mostrou-se superior aos demais em todas as métricas estatísticas, com exceção da mediana, critério que o algoritmo Jaya original foi melhor. LJA teve um desempenho fraco e equiparado com DE, o único quesito que LJA não se mostrou pior que os outros dois algoritmos foi na máxima, quesito que o Jaya original teve um valor maior, ambos com valores muito acima do esperado ultrapassando a casa das centenas.

A Figura 4.16Figura 4.13 representa as conformações da proteína, 2N35 com a menor energia livre encontrada para cada otimização. Assim como pra quinta

proteína, nas conformações de Jaya, EO-Jaya, ABC e GWO, é claramente notável a semelhança entre as dobras das conformações e das três α-hélices encontrada na proteína original de acordo com a Figura 3.1, o que não ocorre nas conformações de LJA e DE, as quais tiveram valores mínimos e de RMSD altos. É também possível observar a formação de núcleos hidrofóbicos, especialmente nas proteínas conformadas pelos algoritmos ABC e GWO.

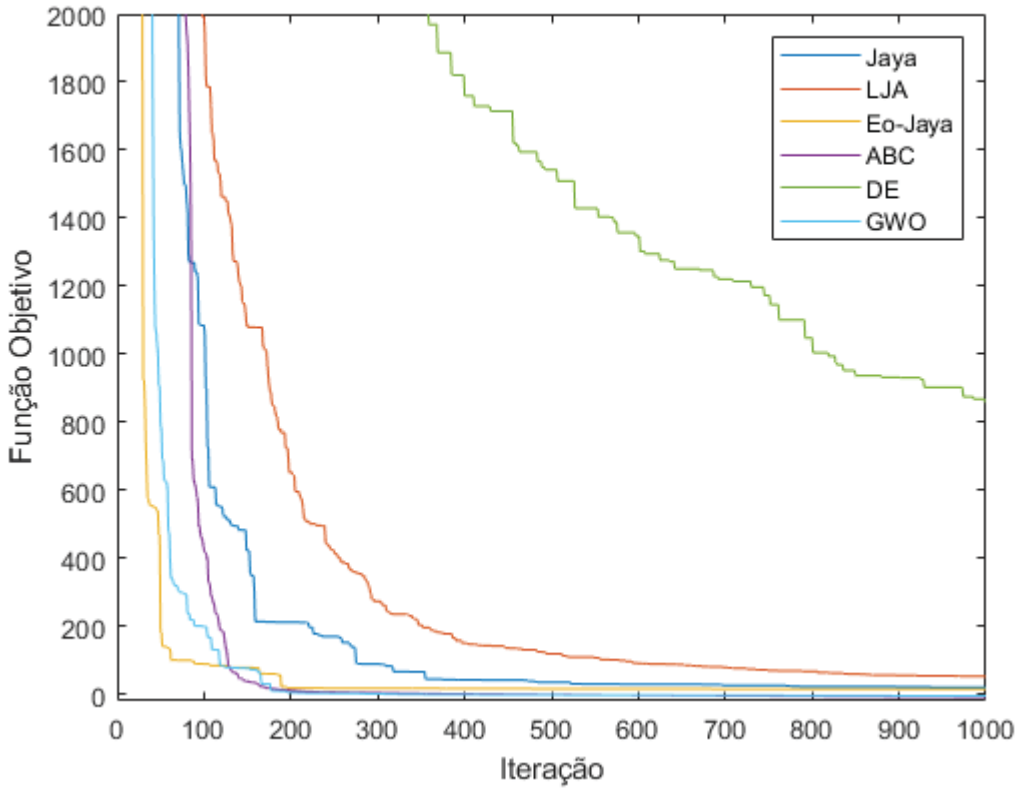
Figura 4.16 - Melhores conformações para a proteína 2N35



Fonte: a autora, 2021.

A Figura 4.17Figura 4.11 apresenta a convergência dos algoritmos de acordo com a média de cada iteração e é possível notar que as otimizações normalmente convergem até 400ª iteração, com exceção da DE, que parece que ainda não se estabiliza pro completo até a última iteração e de LJA que demora além da 600ª iteração para se estabilizar. Assim como para a proteína 3V1A, é notável a semelhança da curva de EO-Jaya com as de ABC e GWO que obtiveram os melhores resultados. Jaya se mostra mediana em relação às outras, com uma curva não tão rápida quando EO-Jaya, mas mais consistente do que LJA e DE.

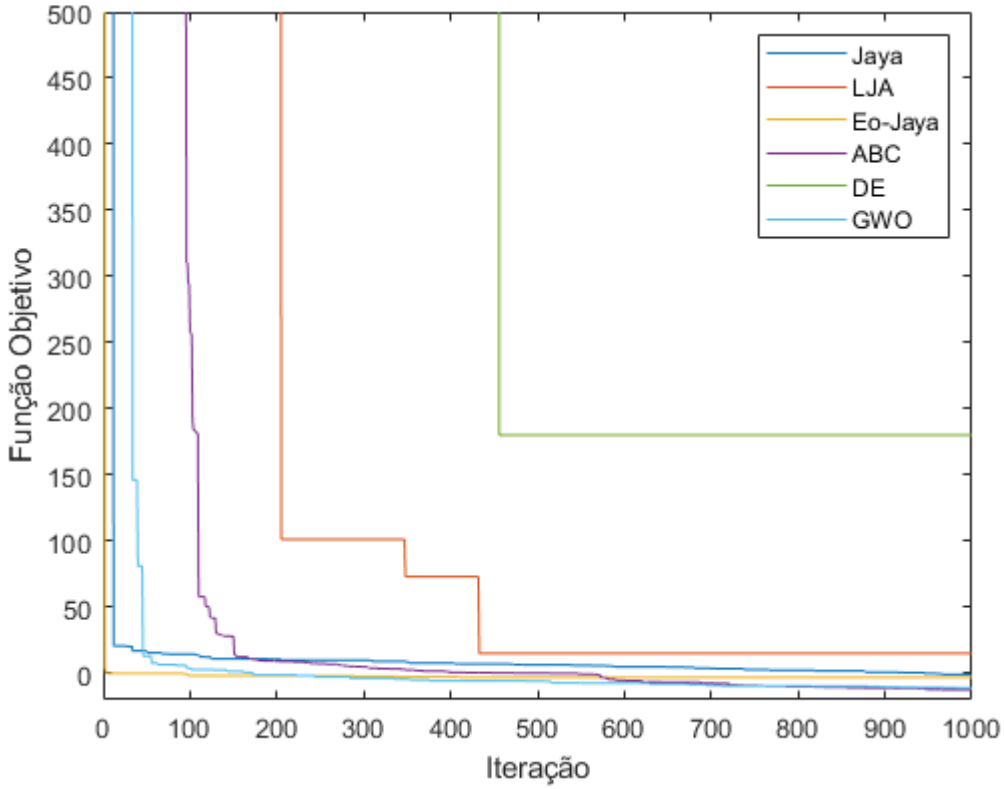
Figura 4.17 - Convergência da média de cada iteração para a proteína2N35



Fonte: a autora, 2021.

A Figura 4.18 mostra o desempenho dos algoritmos com seus melhores resultados. EO-Jaya e GWO e Jaya se mostraram muito parecidos, enquanto ABC que teve o melhor resultado, demorou mais para convergir. DE novamente converge-se em uma curva bastante brusca, ou seja, de uma geração para outra, essa otimização encontra um valor muito menor e se mantem nele até o final das iterações, dando a ideia de ser mais factível a ficar presa em mínimos locais, sendo ainda a otimização que demorou mais para encontrar um bom valor. LJA também é composta por degraus apurados como DE, porém se converge mais rápido e em um valor menor. Nota-se que EO-Jaya foi a otimização que convergiu mais rápido e com um valor comparável a GWO e ABC, sendo novamente o melhor algoritmo entre os que são baseados em Jaya.

Figura 4.18 - Convergência do melhor resultado de cada algoritmo para a proteína 2N35



Fonte: a autora, 2021.

4.7 PROTEÍNA (7) – 1CB9

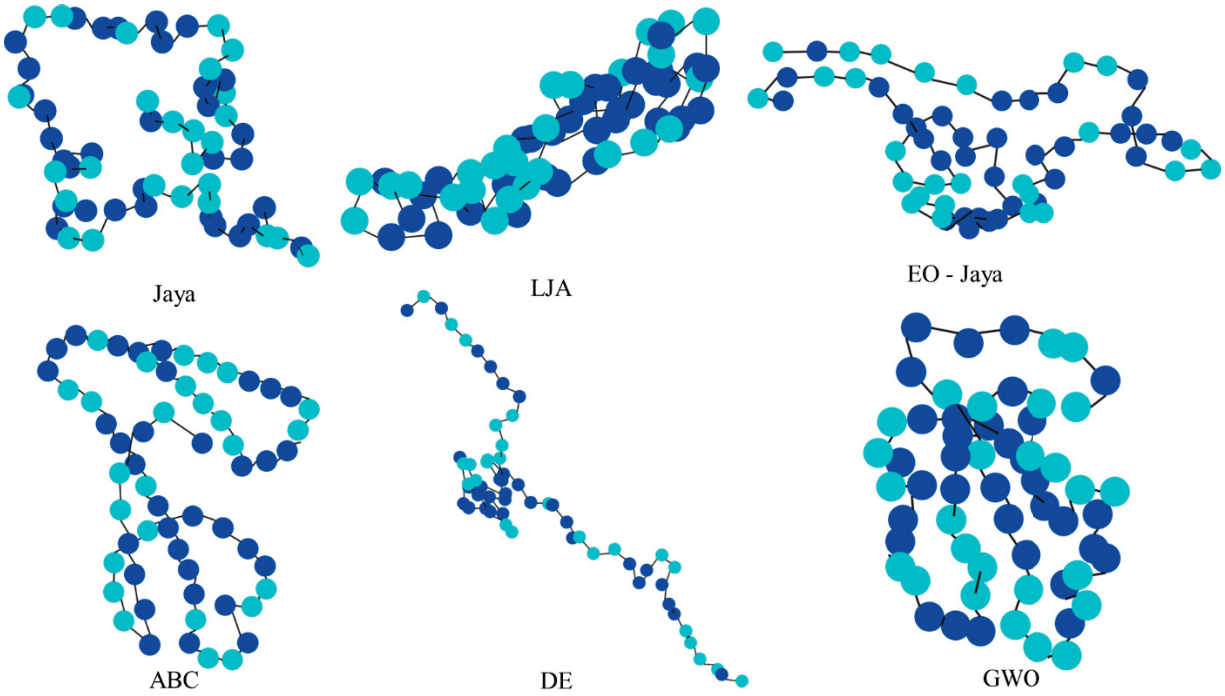
Na Tabela 4.1, é possível visualizar que o algoritmo EO-Jaya foi o melhor na métrica de mínimo, e por consequência em RMSD também. ABC foi o melhor nas demais métricas, seguido por perto pelo GWO. DE foi claramente o pior algoritmo em todas as avaliações, tendo valores acima de 100 para o valor mínimo e acima de 10000 para valor máximo. Nenhum dos algoritmos teve valor RMSD dentro do padrão, mas não muito maior, com exceção de DE que teve valor perto de 150, quase trinta vezes maior que o esperado.

Entre os algoritmos baseados em Jaya, EO-Jaya mostrou-se superior aos demais em todas as métricas estatísticas, sendo o único que não mostrou valores acima de mil na métrica máxima. LJA foi novamente o pior dos três em todas as métricas, sendo apenas ultrapassado pela máxima de Jaya que foi levemente maior.

Pela Figura 4.19 é possível notar que as conformações começam a se diferenciar cada vez umas das outras. Contudo a proteína conformada pelo algoritmo DE se mostra bem mais distinta das demais, sem mostrar a ideia das β-

folhas que a proteína original tem, diferente das demais que essa ideia é expressa pelas dobras formadas. Ainda é possível notar os núcleos apolares se formando, especialmente na conformação da otimização EO-Jaya, ABC e GWO, as com melhor rendimento. LJA também expressa formações de núcleo hidrofóbicos em seu interior.

Figura 4.19 - Melhores conformações para a proteína 1CB9



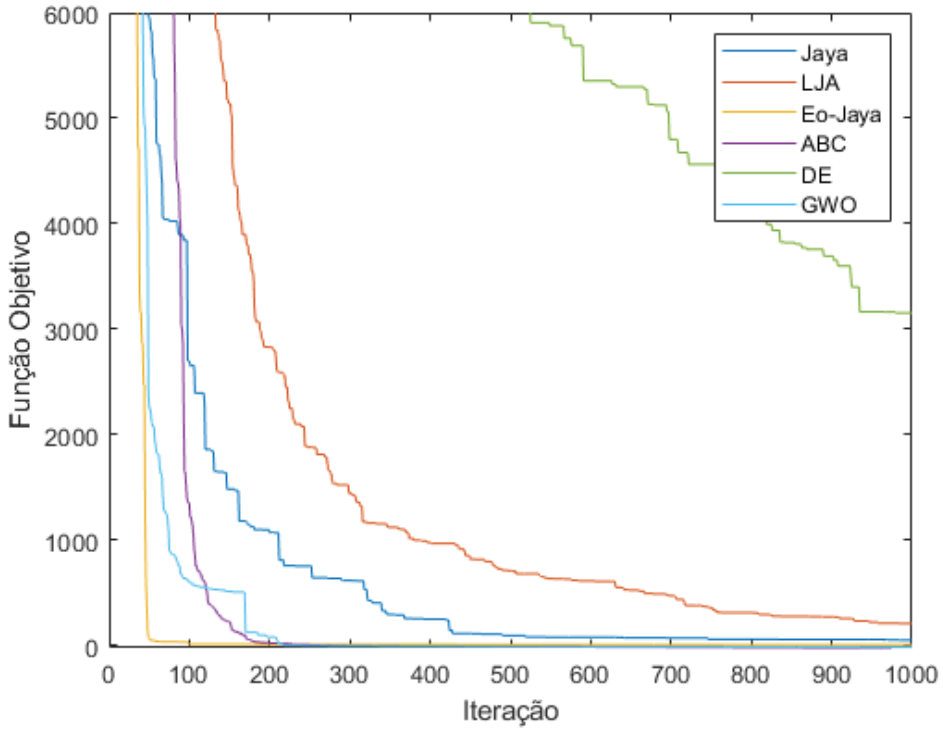
Fonte: a autora, 2021.

A Figura 4.20Figura 4.11 apresenta a convergência dos algoritmos de acordo com a média de cada iteração e é possível notar que a maioria das otimizações convergem até 500ª iteração, entretanto LJA tem uma curva mais lenta e não parece se estabilizar perfeitamente até a 1000ª iteração, já DE obteve a pior curva, a mais lenta. EO-Jaya se estabiliza por primeiro, sendo assim a mais rápida de todas as otimizações.

A Figura 4.21 mostra o desempenho dos algoritmos com seus melhores resultados. EO-Jaya e Jaya são os primeiros a convergir, encontrando o valor ótimo mais rapidamente. GWO e ABC tem curvas bastante semelhantes e convergem mais devagar que EO-Jaya e Jaya. LJA tem a pior curva entre as otimizações

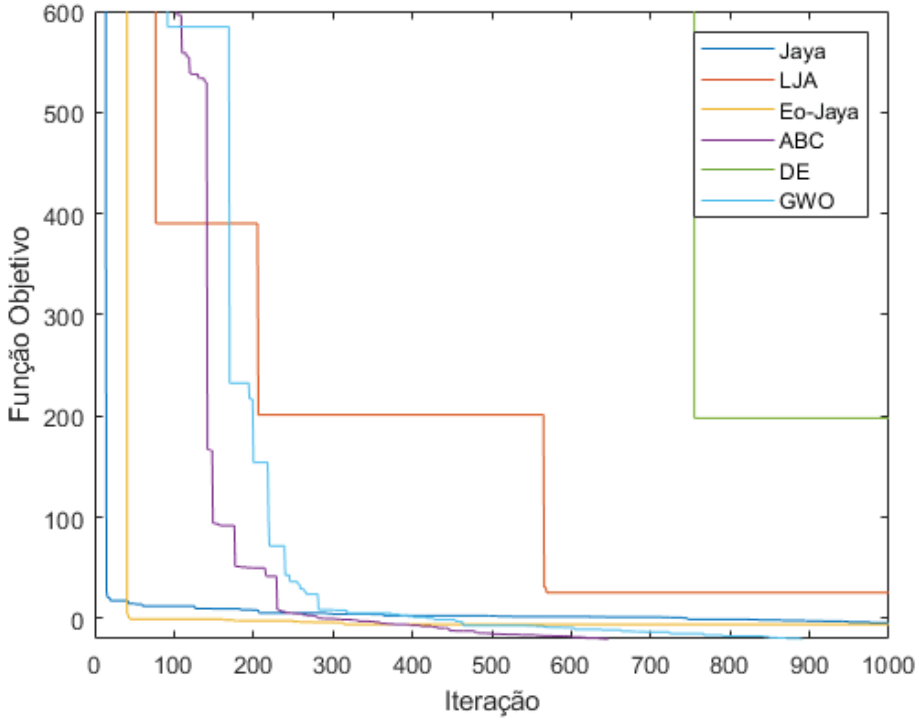
baseadas em Jaya, sendo lenta e se estabilizando em um valor relativamente alto. DE teve o comportamento mais insatisfatório e lento.

Figura 4.20 - Convergência da média de cada iteração para a proteína 1CB9



Fonte: a autora, 2021.

Figura 4.21 - Convergência do melhor resultado de cada algoritmo para a proteína 1CB9



Fonte: a autora, 2021.

4.8 PROTEÍNA (8) – 1PTF

Como é possível ver na Tabela 4.1, EO-Jaya teve o melhor resultado mínimo, assim como menor valor RMSD, enquanto ABC teve os melhores valores nas demais métricas. GWO teve resultados parecidos com ABC. DE foi claramente o pior algoritmo em todas as avaliações, tendo valores acima de dezenas de milhares em todos os critérios. Os algoritmos obtiveram valores de RMSD acima de 5Å, alguns como GWO, ABC, Jaya e EO-Jaya obtiveram valores ainda aceitáveis, enquanto DE e LJA obtiveram valores RMSD nada informativos a respeito da conformação certa da proteína.

Entre os algoritmos baseados em Jaya, EO-Jaya mostrou-se superior aos demais em todas as métricas estatísticas, especialmente nos quesitos de mínimo e mediana, o qual obteve valores condizentes com GWO e ABC, senão melhores. LJA foi novamente menos satisfatório dos três em todas as métricas, sendo apenas ultrapassado pela máxima de Jaya que foi levemente maior.

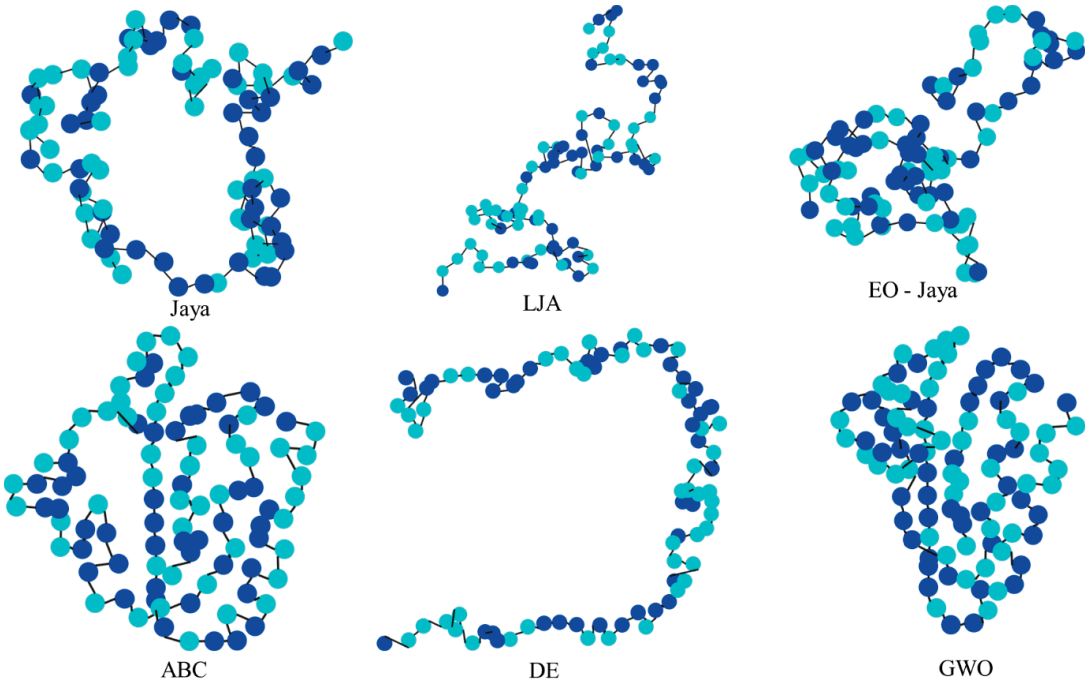
Pela Figura 4.22 nota-se que as conformações estão bastante distintas umas das outras. Ainda é possível ver certa similaridade entre as proteínas conformadas por EO-Jaya, GWO e ABC. Contudo as proteínas conformadas pelos algoritmos DE e LJA se mostram bem mais distinta das demais, sem mostrar os enovelamentos que a proteína original tem. Ainda é possível notar os núcleos apolares se formando, especialmente na conformação da otimização Jaya, EO-Jaya, ABC e GWO, as com melhor rendimento.

A Figura 4.23 apresenta a convergência dos algoritmos de acordo com a média de cada iteração e é possível notar que a maioria das otimizações convergem até 600ª iteração, entretanto LJA tem uma curva lenta e não parece se estabilizar perfeitamente até a 1000ª iteração, assim como DE, que obteve a curva mais lenta. Novamente, EO-Jaya se estabiliza por primeiro, sendo assim a mais rápida de todas as otimizações. GWO e Jaya obtiveram curvas médias muito semelhantes, descrevendo bons resultados em ambas metaheurísticas.

A Figura 4.24 mostra o desempenho dos algoritmos com seus melhores resultados. EO-Jaya e Jaya são os primeiros a convergir, encontrando o valor ótimo mais rapidamente. GWO e ABC e LJA tem curvas bastante semelhantes e convergem mais devagar que EO-Jaya e Jaya. DE novamente teve o

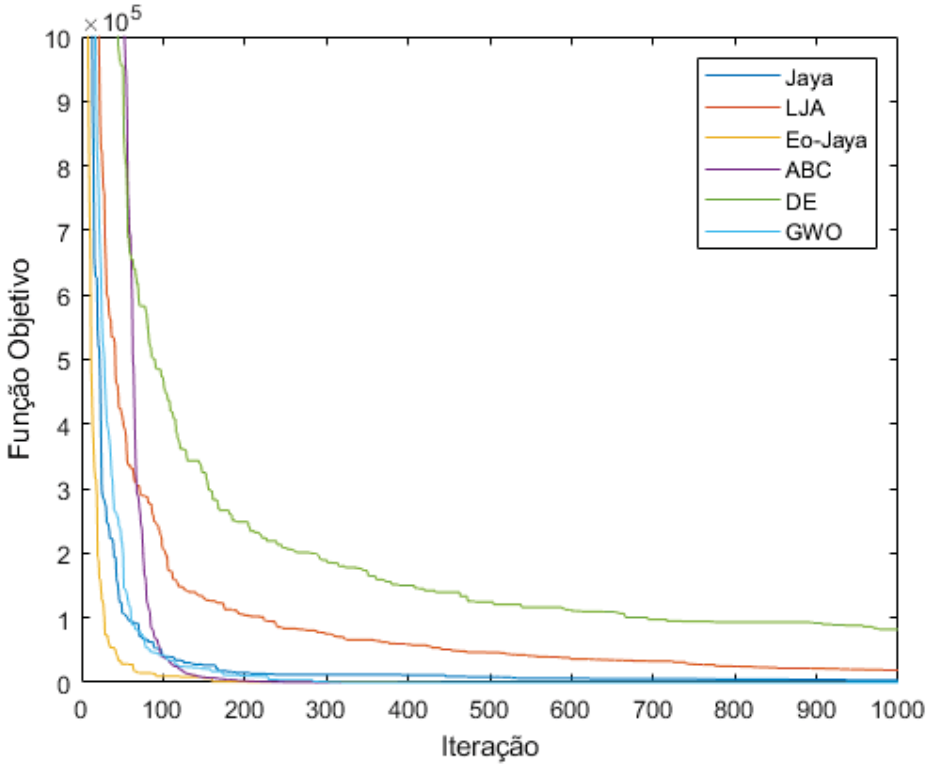
comportamento mais insatisfatório e lento, se mantendo em valores altos por muito tempo.

Figura 4.22 - Melhores conformações para a proteína 1PTF



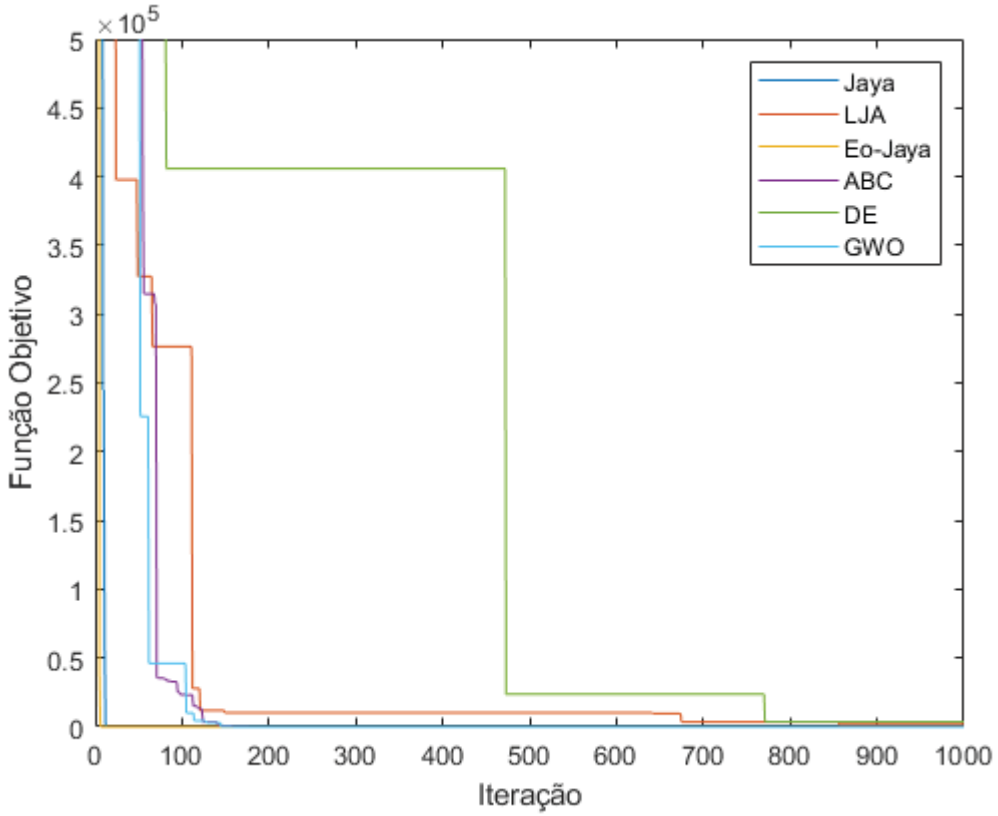
Fonte: a autora, 2021.

Figura 4.23 - Convergência da média de cada iteração para a proteína 1PTF



Fonte: a autora, 2021.

Figura 4.24 - Convergência do melhor resultado de cada algoritmo para a proteína 1PTF



Fonte: a autora, 2021.

4.9 PROTEÍNA (9) – 2G13

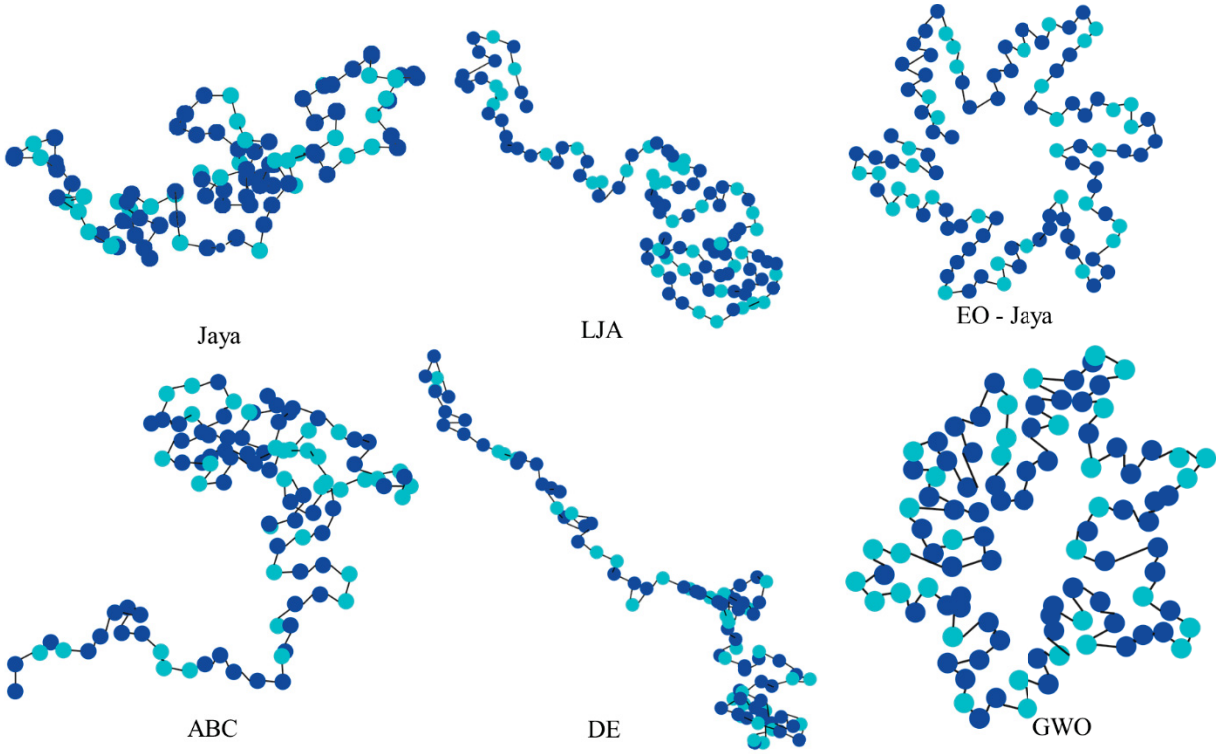
Como é possível ver na Tabela 4.1, Eo-Jaya teve melhor mínimo e valor RMSD, entanto ABC foi a otimização com melhor resultado em média, mediana e máxima. Na questão do desvio padrão, GWO teve um valor melhor. DE foi claramente o pior algoritmo em todas as avaliações, tendo valores acima de dezenas de milhares em todos os critérios. Todos os algoritmos obtiveram valores de RMSD acima de 5Å, mesmo assim, DE e LJA obtiveram valores completamente não informativos a respeito das conformações, sendo mais de mil vezes maior que o valor padrão.

Entre os algoritmos baseados em Jaya, EO-Jaya mostrou-se superior aos demais em todas as métricas estatísticas, sendo igualável às otimizações ABC e GWO. LJA teve o pior desempenho entre as três metaheurísticas, mostrando valores muito altos e inconsistentes. Jaya mesmo tendo valores coerentes de mínima e mediana, suas respostas de máxima, média e mediana foram similares ao LJA,

demostrando que essa otimização não possui uma boa repetibilidade quando se trata de proteínas grandes.

Na Figura 4.25 observa-se a disparidade entre as conformações, embora certa semelhança entre a conformação de GWO e EO-Jaya seja visível. ABC e Jaya também demonstram certas dobras em comum. Já as proteínas conformadas pelos algoritmos DE e LJA se mostram bem mais distinta das demais, sem mostrar os enovelamentos que a proteína original tem. Ainda é possível notar os núcleos apolares se formando ao centro da proteína, especialmente na conformação da otimização Jaya, EO-Jaya, ABC e GWO.

Figura 4.25 - Melhores conformações para a proteína 2G13

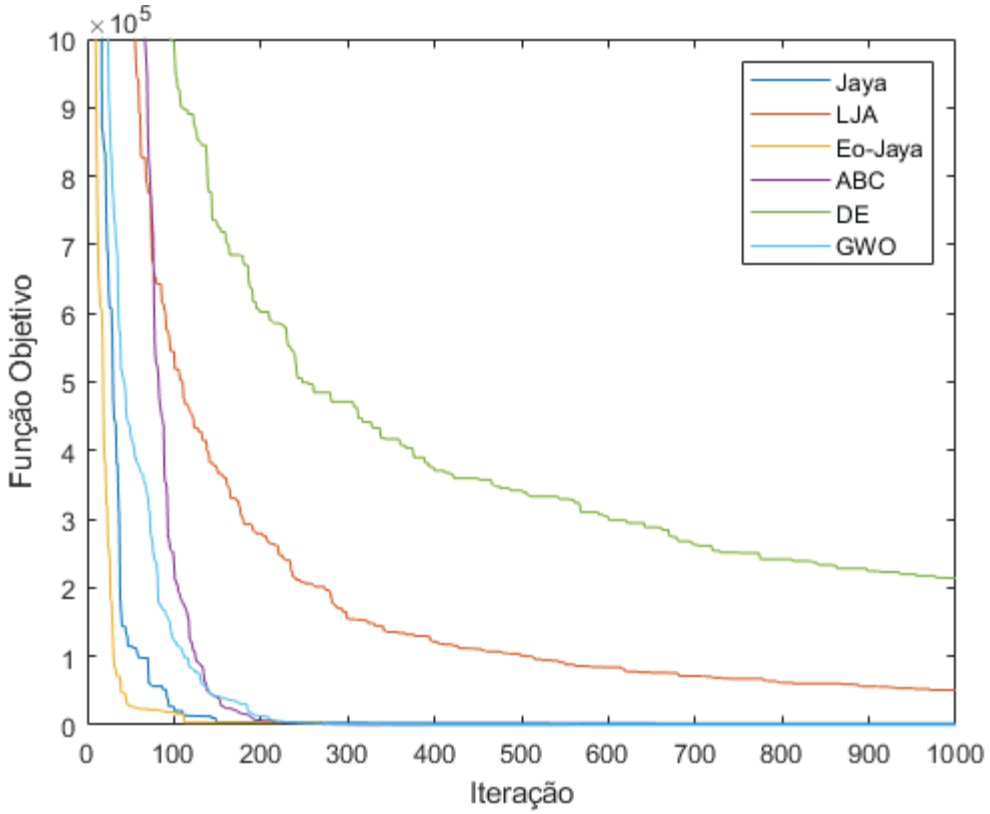


Fonte: a autora, 2021.

A Figura 4.26 apresenta a convergência dos algoritmos de acordo com a média de cada iteração e é possível notar que com exceção de DE e LJA os algoritmos se convergem rapidamente. DE e LJA além de demorarem mais para convergir, se estabilizam em valores demasiados altos. Novamente, EO-Jaya se estabiliza por primeiro, sendo assim a mais rápida de todas as otimizações, seguida por Jaya que também converge em um tempo satisfatório. ABC e GWO apesar de terem os melhores resultados de média, não foram tão rápidos para se estabilizar.

LJA e DE aparentam não ter se estabilizado até a 1000ª iteração, possuindo ainda valores altos.

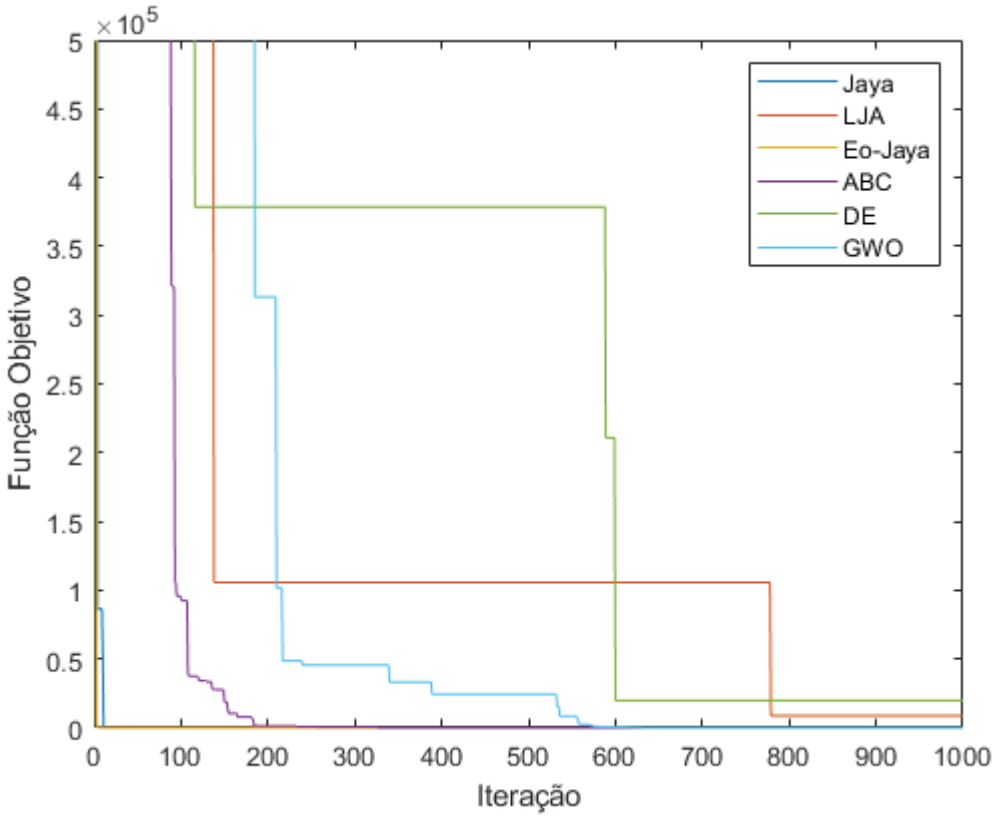
Figura 4.26 - Convergência da média de cada iteração para a proteína 2G13



Fonte: a autora, 2021.

A Figura 4.27 mostra o desempenho dos algoritmos com seus melhores resultados. EO-Jaya e Jaya são os primeiros a convergir, encontrando o valor ótimo mais rapidamente, antes da 100ª iteração. ABC e GWO convergiram mais lentamente que as duas melhores otimizações baseadas em Jaya, convergindo antes da 200ª e da 600ª iteração, respectivamente. Enquanto, DE e LJA demonstraram o comportamento mais insatisfatório e lento, se mantendo em valores altos por muito tempo. LJA encontra um valor menor que DE, contudo demora mais para isso, se mantendo por quase 700 iterações no mesmo mínimo local.

Figura 4.27 - Convergência do melhor resultado de cada algoritmo para a proteína 2G13



Fonte: a autora, 2021.

4.10 PROTEÍNA (10) – 3F00

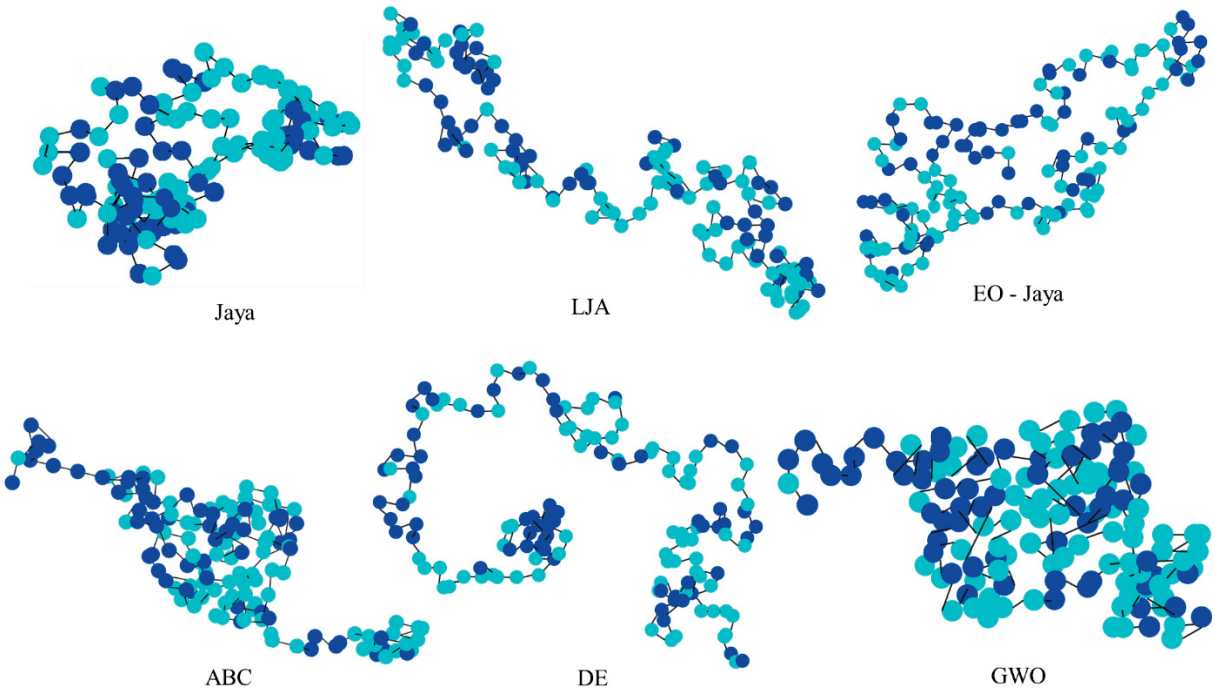
A décima proteína é a maior de todas e a única com mais de 100 aminoácidos. Na Tabela 4.1, a otimização ABC teve a melhor métrica de mínimo e consequentemente o melhor RMSD, contudo GWO teve um valor bastante parecido nesses dois critérios e foi melhor nos demais quesitos, EO-jaya teve valor de mínimo e RMSD parecidos com GWO. DE foi novamente o pior algoritmo em todas as avaliações, tendo valores acima de centenas de milhares em todos os critérios. Todos os algoritmos obtiveram valores de RMSD acima de 5Å, contudo os valores de EO-Jaya, Jaya, ABC e GWO foram apenas um pouco maiores e ainda demonstram certa compreensão do problema, já DE e LJA obtiveram valores grandes o suficiente para serem considerados totalmente incoerentes com o problema.

Entre os algoritmos baseados em Jaya, EO-Jaya mostrou-se superior aos demais em todas as métricas estatísticas, tendo os valores de mínimo e mediana parecidos com as melhores otimizações ABC e GWO, entretanto seus valores de

máximo e desvio padrão foram altos, mostrando que em proteínas acima de 100 aminoácidos, EO-Jaya começa a perder um pouco de seu padrão de confiança em repetibilidade. LJA teve o pior desempenho entre as três metaheurísticas, mostrando valores muito altos e inconsistentes, parecidos com o DE. Jaya mesmo tendo valores coerentes de mínima, mediana e RMSD, teve respostas de máxima, média e mediana bastante insatisfatórias.

Na Figura 4.28 é possível notar que as conformações têm quase nenhuma semelhança umas com as outras, exceção disso é uma leve afinidade entre as conformações das otimizações ABC, GWO e Jaya. Essa é a maior proteína e sendo a mais complexa, assim, é muito incomum encontrar conformações parecidas. LJA e DE novamente não apresentam nenhuma semelhança com a proteína original, a qual é formada por várias folhas-β, as conformações desses algoritmos aparentam ser apenas uma continuidade quase linear de aminoácidos, sem os dobramentos esperados. Ainda é possível notar os núcleos apolares se formando ao centro da proteína, especialmente na conformação da otimização Jaya, EO-Jaya, ABC e GWO.

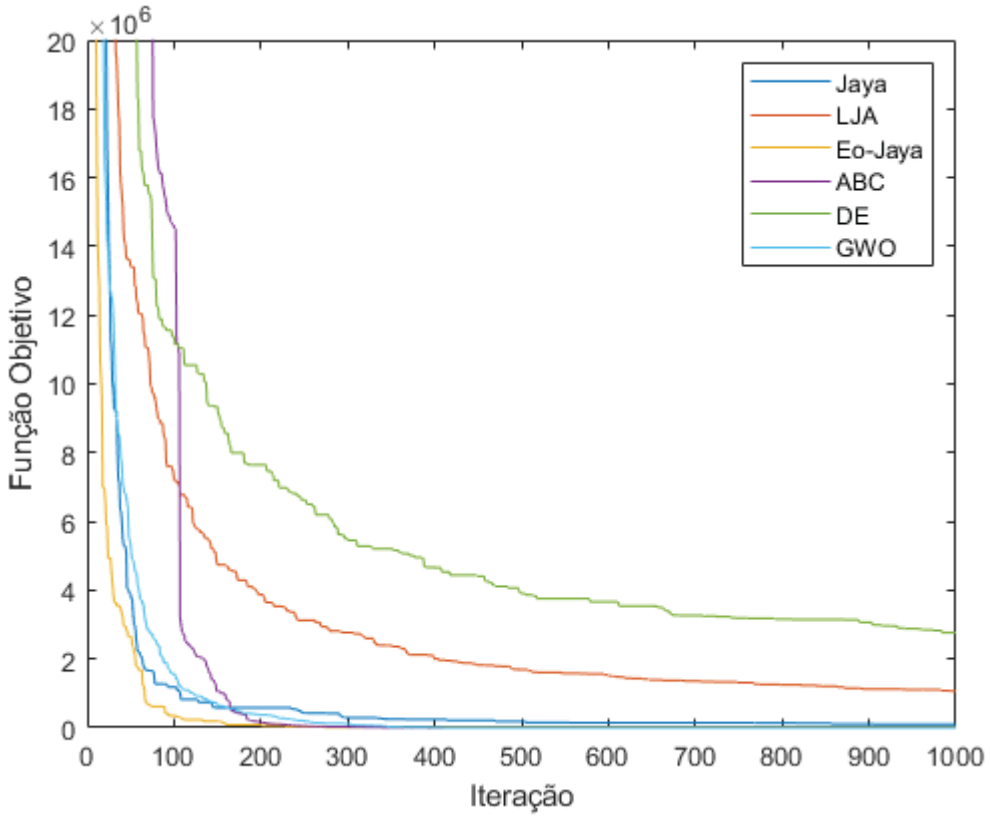
Figura 4.28 - Melhores conformações para a proteína 3F00



Fonte: a autora, 2021.

A Figura 4.29 apresenta a convergência dos algoritmos de acordo com a média de cada iteração e é possível notar que novamente com exceção de DE e LJA os algoritmos se convergem rapidamente. DE e LJA além de demorarem mais para convergir, se estabilizam em valores demasiados altos. Assim como nas proteínas anteriores, EO-Jaya se estabiliza por primeiro, sendo assim a mais rápida de todas as otimizações, seguida por Jaya que também converge em um tempo satisfatório. GWO tem curva parecida com Jaya, também sendo uma das otimizações mais rápidas para convergir. LJA e DE aparentam não ter se estabilizado até a 1000ª iteração, possuindo ainda valores altos, contudo é aparente que esses algoritmos se estabilizam melhor nessa proteína do que nas anteriores (8ª e 9ª), o que é contraintuitivo.

Figura 4.29 - Convergência da média de cada iteração para a proteína 3F00

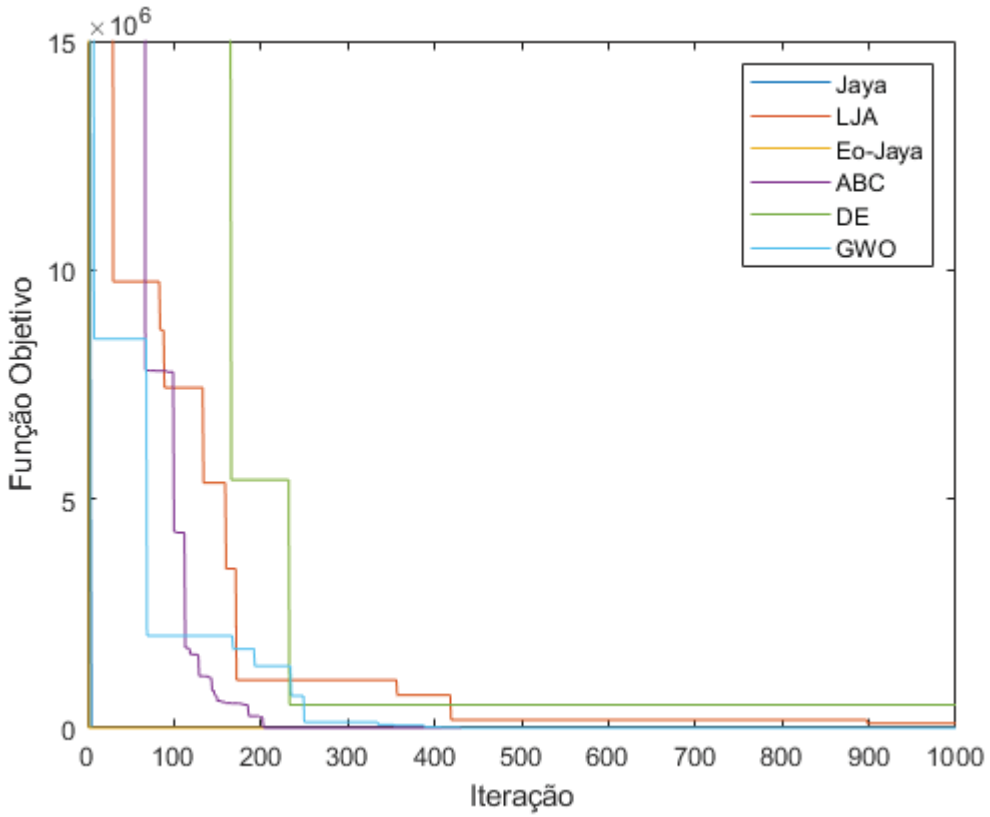


Fonte: a autora, 2021.

A Figura 4.30 mostra o desempenho dos algoritmos com seus melhores resultados. EO-Jaya e Jaya são os primeiros a convergir, encontrando o valor ótimo mais rapidamente, antes da 100ª iteração. As demais otimizações parecem

convergir numa velocidade mais parelha do que nas proteínas vistas anteriormente. Surpreendentemente, DE parece se estabilizar antes de GWO, embora com um valor quase 10^6 vezes maior. ABC é a terceira a convergir, então além de se estabilizar rapidamente, faz isso em um valor aceitável. LJA é a última a convergir, embora faça isso em um ponto de valor menor que DE.

Figura 4.30 - Convergência do melhor resultado de cada algoritmo para a proteína 3F00



Fonte: a autora, 2021.

4.11 TESTE DE WILCOXON

O teste de significância estatística não paramétrica de Wilcoxon tem como objetivo comparar as performances de cada algoritmo no sentido de verificar se existem diferenças significativas entre os seus resultados em cada situação. A hipótese nula H_0 afirma que o erro médio da amostra do método de controle é igual ou maior do que os outros algoritmos comparados. Consequentemente, a hipótese H_1 significa que o erro mediano do método de controle é menor do que o outro algoritmo. Assim a Tabela 4.2 mostra os valores referentes ao teste de Wilcoxon

para cada proteína, sendo as três otimizações baseadas em Jaya usadas como algoritmos de controle.

Tabela 4.2 - Teste de Wilcoxon dos melhores resultados em 50 rodadas

Proteína	Algoritmo	Controle: Jaya		Controle: LJA		Controle: EO-Jaya	
		Valor *p*	Rejeitar H_0?	Valor *p*	Rejeitar H_0?	Valor *p*	Rejeitar H_0?
(1) -1CB3	Jaya	-	-	2.4204e-10	Sim	2.0452e-09	Sim
	LJA	2.4204e-10	Sim	-	-	0.6270	Não
	EO-Jaya	2.0452e-09	Sim	0.6270	Não	-	-
	ABC	8.9852e-18	Sim	5.3319e-16	Sim	3.5664e-15	Sim
	DE	0.1526	Não	8.5886e-06	Sim	3.2739e-05	Sim
	GWO	5.8408e-12	Sim	0.0248	Sim	0.0235	Sim
(2) - 5HPP	Jaya	-	-	7.0661e-18	Sim	3.9657e-17	Sim
	LJA	7.0661e-18	Sim	-	-	8.7103e-07	Sim
	EO-Jaya	3.9657e-17	Sim	8.7103e-07	Sim	-	-
	ABC	7.0661e-18	Sim	7.0661e-18	Sim	7.0661e-18	Sim
	DE	6.2019e-11	Sim	8.9852e-18	Sim	6.8803e-14	Sim
	GWO	7.0661e-18	Sim	7.5041e-18	Sim	1.0754e-17	Sim
(3) – 1EDN	Jaya	-	-	8.4620e-18	Sim	3.1940e-15	Sim
	LJA	8.4620e-18	Sim	-	-	1.4288e-13	Sim
	EO-Jaya	3.1940e-15	Sim	1.4288e-13	Sim	-	-
	ABC	7.0661e-18	Sim	7.0661e-18	Sim	7.0661e-18	Sim
	DE	0.0104	Sim	8.3791e-16	Sim	2.6913e-16	Sim
	GWO	7.0661e-18	Sim	7.0661e-18	Sim	4.2485e-16	Sim
(4) - 1GK7	Jaya	-	-	2.0707e-06	Sim	0.0704	Não
	LJA	2.0707e-06	Sim	-	-	0.0207	Sim
	EO-Jaya	0.0704	Não	0.0207	Sim	-	-
	ABC	7.0661e-18	Sim	7.0661e-18	Sim	7.0661e-18	Sim
	DE	1.6998e-16	Sim	7.9688e-18	Sim	8.3791e-16	Sim
	GWO	7.0661e-18	Sim	7.0661e-18	Sim	2.0710e-17	Sim
(5) - 3V1A	Jaya	-	-	1.0746e-10	Sim	0.3259	Não
	LJA	1.0746e-10	Sim	-	-	3.9197e-05	Sim
	EO-Jaya	0.3259	Não	3.9197e-05	Sim	-	-
	ABC	7.0661e-18	Sim	7.0661e-18	Sim	7.0661e-18	Sim
	DE	7.5510e-17	Sim	1.4496e-17	Sim	7.0661e-18	Sim
	GWO	7.0661e-18	Sim	7.0661e-18	Sim	8.9852e-18	Sim
(6) - 2N35	Jaya	-	-	8.5534e-11	Sim	0.8496	Não
	LJA	8.5534e-11	Sim	-	-	2.8105e-12	Sim
	EO-Jaya	0.8496	Não	2.8105e-12	Sim	-	-
	ABC	7.0661e-18	Sim	7.0661e-18	Sim	7.0661e-18	Sim
	DE	8.4620e-18	Sim	7.0661e-18	Sim	7.0661e-18	Sim
	GWO	7.0661e-18	Sim	7.0661e-18	Sim	5.9784e-17	Sim
(7) - 1CB9	Jaya	-	-	1.4783e-12	Sim	0.0094	Sim
	LJA	1.4783e-12	Sim	-	-	2.4739e-17	Sim
	EO-Jaya	0.0094	Sim	2.4739e-17	Sim	-	-
	ABC	7.0661e-18	Sim	7.0661e-18	Sim	7.0661e-18	Sim

	DE	1.2866e-17	Sim	4.2058e-17	Sim	7.0661e-18	Sim
	GWO	7.5041e-18	Sim	7.0661e-18	Sim	1.6330e-17	Sim
(8) - 1PTF	Jaya	-	-	1.7284e-14	Sim	0.0736	Não
	LJA	1.7284e-14	Sim	-	-	7.9688e-18	Sim
	EO-Jaya	0.0736	Não	7.9688e-18	Sim	-	-
	ABC	7.0661e-18	Sim	7.0661e-18	Sim	7.0661e-18	Sim
	DE	9.5300e-17	Sim	4.8955e-13	Sim	7.0661e-18	Sim
	GWO	7.0661e-18	Sim	7.0661e-18	Sim	7.0661e-18	Sim
(9) - 2G13	Jaya	-	-	1.2019e-16	Sim	0.5649	Não
	LJA	1.2019e-16	Sim	-	-	7.0661e-18	Sim
	EO-Jaya	0.5649	Não	7.0661e-18	Sim	-	-
	ABC	7.0661e-18	Sim	7.0661e-18	Sim	7.0661e-18	Sim
	DE	9.5403e-18	Sim	1.4066e-12	Sim	7.0661e-18	Sim
	GWO	7.0661e-18	Sim	7.0661e-18	Sim	7.0661e-18	Sim
(10) - 3F00	Jaya	-	-	4.2066e-15	Sim	0.0135	Sim
	LJA	4.2066e-15	Sim	-	-	1.2120e-17	Sim
	EO-Jaya	0.0135	Sim	1.2120e-17	Sim	-	-
	ABC	8.0040e-17	Sim	7.0661e-18	Sim	4.9188e-10	Sim
	DE	2.9538e-17	Sim	9.0136e-12	Sim	7.0661e-18	Sim
	GWO	7.0661e-18	Sim	7.0661e-18	Sim	1.4496e-17	Sim

4.12 DISCUSSÃO DOS RESULTADOS

É notável que quanto menor a proteína, mais preciso os algoritmos se tornam. Isso já era esperado, uma vez que a dimensão do problema está diretamente relacionada com o número de aminoácidos que compõem a proteína. Para proteínas até aproximadamente 50 aminoácidos, ou seja, para otimizações de até 100 dimensões, todos os algoritmos parecem se comportarem de maneira equiparada e coerente. Apesar de alguns valores RMSD serem acida de 5Å, não foram valores muito maiores, demonstrando a capacidade dos algoritmos de cumprirem com a proposta para esse tamanho de dimensão.

Para dimensões maiores as otimizações começaram a perder sua precisão. DE mostrou os piores resultados na maioria dos casos, com valores extremamente altos e nada informativos a respeito da proteína. ABC e GWO obtiveram sempre os melhores valores, independente do tamanho do problema, essas otimizações conseguiram manter um padrão satisfatório semelhante.

A partir da proteína de número 4 LJA passou a ter resultados piores que o algoritmo de Jaya original, mostrando que a variante com voo de Lévy só é uma boa opção ao algoritmo original quando se referem a proteínas pequenas – de dimensão

de até 70, ou 38 aminoácidos. Para proteínas maiores seus resultados foram muito altos, similares aos de DE, sendo assim imprecisos e não conseguem demonstrar com a clareza necessária a conformação proteica.

EO-Jaya por outro lado se mostrou promissora em todas as métricas. Mesmo que seu RMSD não tenha sido tão bom quanto o dos algoritmos ABC e GWO, foi um valor próximo. Além disso, o tamanho da proteína não influenciou muito no desempenho do algoritmo, mantendo sempre um bom padrão. EO-Jaya também foi o na maioria das vezes a otimização mais rápida a convergir, mostrando ser rápida e precisa. Para todos os testes teve resultados melhores que o algoritmo de Jaya original.

A otimização Jaya original obteve valores adequados. Não teve valores absurdos como o caso de LJA e DE, e não obteve valores melhores que os demais algoritmos. Apesar de que em problemas de dimensões maiores que 100, começou a obter valores de máxima, média e mediana acima de mil, o que é considerado, para esse caso, valores muito altos e não informativos. Mesmo assim se mostrou mais capaz de evitar mínimos locais do que sua variante LJA.

Os resultados de RMSD acima de 5Å podem ser explicados pela falta de outras informações essenciais utilizadas na conformação de proteínas, como por exemplo, dados relacionados aos dobramentos das α-hélices e folhas-β. Além disso, o modelo AB *off-lattice* utiliza apenas do conhecimento da hidro afinidade dos aminoácidos e a função de energia reduz o processo de conformação a criação de um núcleo hidrofóbico compacto, com a finalidade de produzir valores menores de energia, o que não reflete verdadeiramente um dobramento real. Dessa forma, a informação do núcleo hidrofóbico é, de fato, um aspecto importante da previsão da estrutura final da proteína, contudo o uso de somente essa informação durante a otimização é insuficiente para uma boa previsão de RMSD quando se utiliza de proteínas grandes. (SILVA e PARPINELLI, 2019).

Apesar desse empecilho, o modelo de rede encontrado fornece certas informações a respeito da estrutura final, podendo ser utilizado para alimentar boas soluções de outros algoritmos de otimização, os quais usam modelos de previsão de estrutura de proteína mais complexos, como por exemplo o modelo de átomo completo (BOIANI e PARPINELLI, 2020).

Quanto maior a proteína maiores os valores da função objetivo durante as primeiras gerações, contudo, todas as proteínas conseguiram chegar a valores

relativamente baixos, garantindo o bom rendimento do algoritmo. A partir da sexta proteína (com exceção da proteína 8), nota-se um desvio padrão grande entre os valores finais da função objetivo durante as cinquenta rodadas do algoritmo, assim, pode-se afirmar que o algoritmo se torna menos conciso à medida que o problema se torna mais complexo.

5 CONSIDERAÇÕES FINAIS

O problema de PSP *ab-initio* é representado por encontrar e representar a estrutura nativa de uma proteína utilizando apenas sua sequência de aminoácidos e a função de energia. Nessa pesquisa, esse problema foi abordado incorporando a modelagem 3D-AB *off-lattice*. A principal característica do desenvolvimento de algoritmos de otimização para abordagem do modelo AB está principalmente na descoberta do núcleo hidrofóbico de proteínas com potencial de mal dobramento e sua identificação.

Sendo o problema de PSP complexo e considerado NP-*hard*, é comum o uso de redes neurais artificiais e metaheurísticas, como o caso de algoritmos evolutivos e algoritmos de inteligência de enxames. Assim, essa pesquisa propôs o uso de uma metaheurística nunca utilizada no problema de PSP *ab-initio*, o algoritmo de otimização Jaya e duas variantes dessa.

Avaliando os resultados obtidos pelos algoritmos, observou-se que, entre as dez proteínas reais escolhidas, para sequências curtas os algoritmos conseguiram atingir valores ótimos. Entretanto, para as sequências mais longas, as conformações obtidas possuem um valor não informativo em termos da métrica RMSD, isso ocorre devido à falta de informações adicionais sobre a conformação da proteína. Embora a formação de um núcleo apolar seja um aspecto importante da previsão da estrutura de uma proteína, o uso de apenas essa característica para a orientação durante a otimização do processo é insuficiente para formar boas previsões RMSD se tratando de proteínas mais complexas.

Nota-se que o algoritmo Jaya consegue cumprir com a proposta, assim como sua variante EO-Jaya, a qual teve resultados excelentes quando comparados às outras metaheurísticas. LJA no entanto só obteve resultados satisfatórios em proteínas de até 38 aminoácidos, a partir desse ponto, seus valores não puderam ser usados como informações uteis a respeito da conformação da proteína real.

A modelagem 3D-AB *off-lattice* ainda pode ser útil para dar algumas sugestões sobre a possível estrutura da proteína ou para servir de entrada para outros algoritmos de otimização mais complexos, como o modelo de átomo completo. A partir de uma análise visual das melhores conformações encontradas pela abordagem proposta, a formação de núcleos hidrofóbicos foi alcançada em todas as sequências de proteínas, indicando que os algoritmos propostos são

adequados para o problema de PSP 3D-AB *off-lattice,* pelo menos para proteínas pequenas. A implementação e disponibilização do algoritmo de otimização Jaya e suas variantes ainda permitirá que novos estudos ou aperfeiçoamentos sejam desenvolvidos nesta área. Assim, os objetivos propostos foram atingidos e os resultados obtidos foram satisfatórios.

5.1 RECOMENDAÇÕES PARA TRABALHOS FUTUROS

Para trabalhos futuros é interessante um estudo da parametrização da variante de Jaya com voo de Lévy. Nessa pesquisa os valores de α e β foram fixos e iguais para todas as proteínas, assim um estudo destinado a variação desses valores para cada proteína é interessante, uma vez que a otimização dessas variáveis é imprescindível para um resultado satisfatório da otimização. Outras recomendações para trabalhos futuros envolvem testar outras variantes ainda de Jaya, incluindo variantes hibridas com outras otimizações. Também é interessante o teste desses algoritmos com outras proteínas, de tamanhos maiores e mais complexas.

REFERÊNCIAS

AL-QURAISHI, M. End-to-end differentiable learning of protein structure. **Cell Systems**, v. 8, n. 4, p. 292 – 301, 2019.

ALSAJRI, M.; ISMAIL, M. A.; SALAMN, S. A. A review on the recent application of Jaya optimization algorithm. In: **1st Annual International Conference on Information and Sciences (AiCIS)**. IEEE, p. 129 – 132, Fallujah, Iraque, 2018.

AMBER Tools for molecular simulations. Disponível em: http://ambermd.org/. Acesso em: 18 mai. 2021.

ANFINSEN, C. B.; SELA, M.; WHITE F. H. Jr. Reductive cleavage of disulfide bridges in ribonuclease. **Science**, v. 125, p. 691 – 692, 1957.

ANFINSEN, C. B. Principles that govern the folding of protein chains. **Science**, v. 181, 223 – 230, 1973.

AYODELE, T. O. Machine Learning Overview. **New Advances in Machine Learning**, v. 1, p. 9 – 19, 2010.

BANDARU, S.; DEB, K. Metaheuristic techniques. **Decision sciences: theory and practice**, v. 220, n. 4598, p. 693 – 750, 2016.

BARTHELEMY, P., BERTOLOTTI, J. e WIERSMA, D. A Lévy flight for light. **Nature**, v. 453, p. 495 – 498, 2008.

BARTUMEUS, F., DA LUZ, M. G. E., VISWANATHAN, G. M. e CATALAN, J. Animal search strategies: a quantitative random-walk analysis. **Ecology,** v. 86, p. 3078–3087, 2005.

BENI, G.; WANG, J. Swarm intelligence in cellular robotic systems. In: Dario, P., Sandini, G.; Aebischer, P. (editores) **Robots and Biological Systems: Towards a New Bionics?**, p. 703 – 712. Springer, Berlin, Alemanha, 1993.

BERGER, B; LEIGHTON, T. Protein Folding in the Hydrophobic-hydrophilic (HP) is NP-complete. **In: Proceedings of the 2nd Annual International Conference on Computational Molecular Biology**, RECOMB'98, ACM, Nova York, p. 30 – 39, Estados Unidos da América,1998.

BOIANI, M.; PARPINELLI, R. S. A GPU-Based Hybrid jDE Algorithm Applied to the 3D-AB Protein Structure Prediction. **Swarm and Evolutionary Computation**, v. 58, 2020.

BOŠKOVIC, B.; BREST, J. Genetic algorithm with advanced mechanisms applied to the protein structure prediction in a hydrophobic-polar model and cubic lattice. **Applied Soft Computing**, v. 45, p. 61-70, 2016.

BOŠKOVIC, B.; BREST, J. Two-phase protein folding optimization on a three-dimensional AB off-lattice model. **Swarm and Evolutionary Computation**, v. 57, 2020.

BRANDEN, C.; TOOZE, J. **Introduction to Protein Structure**. 2ª edição. New York: Garland Publishing, 1999.

BROCKMANN, D., HUFNAGEL, L. e GEISEL, T. The scaling laws of human travel. **Nature**, v. 439, p. 462–465, 2006.

BRYSON, M. Heavy Tailed Distributions: Properties and Tests. **Technometrics**, v. 16, n. 1, p. 61-68, 1974.

CARRASCO, J.; GARCÍA, S.; RUEDA, M M.; DAS, S; HERRERA, F. Recent trends in the use of statistical tests for comparing swarm and evolutionary computing algorithms: Practical guidelines and a critical review**. Swarm and Evolutionary Computation**, v. 54, article number 100665, 2020.

CHEN, X.; SONG, S.; JI, J.; TANG, Z.; TODO, Y. Incorporating a multiobjective knowledge-based energy function into differential evolution for protein structure prediction. **Information sciences**, v. 540, p. 69-88, 2020.

CHENG-YUAN, L.; YAN-RUI, D.; WEN-BO, X. Multiple-layer quantum-behaved particle swarm optimization and toy model for protein structure prediction. **In: 2010 Ninth International Symposium on Distributed Computing and Applications to Business, Engineering and Science**, Hong Kong, p. 92–96, 2010.

CHARMM Chemistry at Harvard Macromolecular Mechanics. Disponível em: https://www.charmm.org/. Acesso em: 18 mai. 2021.

Chong, K. L., Lai, S. H., Ahmed, A. N., Jaafar, W. Z. W., & El-Shafie, A. Optimization of hydropower reservoir operation based on hedging policy using Jaya algorithm. **Applied Soft Computing**, v. 106, número de artigo: 107325, 2021.

CONTESSOTO, V. G.; OLIVEIRA, A. B. JR.; CHAHINE, J.; LEITE, V. B. P. Introdução ao problema de enovelamento de proteínas: uma abordagem utilizando modelos computacionais simplificados. **Revista Brasileira de Ensino de Física**, v. 40, n. 4, 2018.

CORRAL, A. Universal earthquake-occurrence jumps, correlations with time, and anomalous diffusion. **Physical Review Letters**, v. 97, article number: 178501, 2006.

DALVI, R. de F. **Suporte a detecção e classificação de câncer a partir de mamografias digitalizadas e redes neurais convolucionais**. Dissertação (Mestrado) – Universidade Federal do Espírito Santo, Vitória, ES, 2018.

DAS, S.; MULLICK, S.S.; SUGANTHAN, P. Recent advances in differential evolution an updated survey. **Swarm and Evolutionary Computation**, v. 27, p. 1 – 30, 2016.

DAWSON, W. M.; RHYS, G. G.; WOOLFSON, D. N. Towards functional de novo designed proteins. **Current Opinion in Chemical Biology**, v. 52, p. 102 – 111, 2019.

DHINGRA, S.; SOWDHAMINI, R.; CADET, F. A glance into the evolution of template-free protein structure prediction methodologies. **Biochimie**, v. 175, p. 85 – 92, 2020.

DILL, K. A. Theory for the folding and stability of globular proteins. **Biochemistry**, v. 24, p. 1501 – 1509, 1985.

DILL, K. A.; BROMBERG, S.; YUE, K.; FIEBIG, K. M.; YEE, D. P.; THOMAS, P. D.; CHAN, H. S. Principles of protein folding – a perspective from simple exact models. **Protein Science**, v. 4, p. 561–602, 1995.

DU, K. L.; SWAMY, M. N. S. **Search and Optimization by Metaheuristics**. Primeira Edição. Birkhäuser: Montreal, Canadá, 2016.

DUAN, Y., KOLLMAN, P.A. Pathways to a protein folding intermediate observed in a 1-microsecond simulation in aqueous solution. **Science**, v. 282, pp. 740 – 744, 1998.

DUBEYA S. P.; KINI, N. G.; KUMAR, M. S.; BALAJI, S. Ab initio protein structure prediction using GPU computing. **Perspectives in Science**, v. 8, p. 645–647, 2016.

EBERHART, R.C.; SHI, Y. Particle swarm optimization: developments, applications and resources. In: **Proceedings of the 2001 Congress on Evolutionary Computation**, v. 1, p. 81 – 86, Seoul, Coréia do Sul, 2001.

EBERHART, R.; KENNEDY, J. A new optimizer using particle swarm theory. **In: Proceedings of the Sixth International Symposium on Micro Machine and Human Science**, p. 39 – 43, Nagoya, Japão, 1995.

ENCAD University of Washington Departments Web Server. Disponível em: http://depts.washington.edu/. Acesso em 18 mai. 2021.

FERINA, J. DAGGETT, V. Visualizing Protein Folding and Unfolding. **Journal of Molecular Biology**, v. 431, n. 8, p. 1540-1564, 2019.

FERREIRA, A. dos S. **Redes Neurais Convolucionais Profundas na Detecção de Plantas Daninhas em Lavoura de Soja.** Dissertação (Mestrado) – Universidade Federal de Mato Grosso do Sul, Mato Grosso do Sul, p. 70, 2017.

GAIKWAD, M. U.; KRISHNAMOORTHY, A.; JATTI, Vi. S. Implementation of Jaya algorithm for process parameter optimization during EDM processing of NiTi 60alloy. **Materials Today**: Proceedings, In press, 2021 .

GAMBELLA, C.; GHADDAR, B.; SAWAYA, J. N. Optimization problems for machine learning: A survey. **European Journal of Operational Research**, v. 290, n. 3, p. 807-828, 2021.

GROMOS. Disponível em: http://www.gromos.net/. Acesso em: 18 mai. 2021.

GUEX, N.; PEITSCH, M. C. Protein structure: comparative protein modelling and visualisation. Disponível em: <http://swissmodel.expasy.org/course/course-index.htm>, 2007.

GINALSKI, K. Comparative Modeling for Protein Structure Prediction. **Current Opinion in Structural Biology**, v. 16, n. 2, p. 172 – 177, 2006.

GOLDBERG, D. E. **Genetic Algorithms in Search, Optimization, and Machine Learning**. Reading, USA: Addison-Wesley, 1989.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep learning**. MIT Press, 2016.

HAYKIN, S. S. **Redes Neurais**. 2ª edição. Bookman, 2001.

HE, K.; ZHANG, X.; REN, S.; SUN, J. Deep residual learning for image recognition. **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**, p. 770–778, San Juan, Estados Unidos da América 2016.

HELLING, R.; LI, H.; TANG, C.; WINGREEN, N. Emergence of preferred structures in a simple model of protein folding. **Science**, v. 273, n.5275, p. 666 – 669, 1996.

HINTON, G. E.; OSINDERO, S.; TEH, Y.-W. A fast learning algorithm for deep belief nets. **Neural computation**, v. 18, n. 7, p. 1527 – 1554, 2006.

HOLLAND, J.H. **Adaptation in Natural and Artificial Systems**. Cambridge, MA: MIT Press, 1975.

IACCA, G.; DOS SANTOS JUNIOR, V. C.; DE MELO, V. V. An improved Jaya optimization algorithm with Levy flight. **Expert Systems with Applications**, v. 165, número do artigo: 113902, 2020.

ICHINOMIYA, T.; OBAYASHI, I.; HIRAOKA Y. Protein-Folding Analysis Using Features Obtainedby Persistent Homology. **Biophysical Journal**, v. 118, p. 2926 – 2937, 2020.

INGLE, K. K.; JATOTH, R. K. An efficient JAYA algorithm with Lévy flight for non-linear channel equalization. **Expert Systems with Applications**, v. 145, article number: 112970, 2019.

JANA, N.; D. DAS, S.; SIL, J. **A Metaheuristic Approach to Protein Structure Prediction: Algorithms and Insights from Fitness Landscape Analysis**. Primeira edição, v.31. Springer, 2018.

JANA, N.; D. DAS, S.; SIL, J. Selection of appropriate metaheuristic algorithms for protein structure prediction in AB off-lattice model: a perspective from fitness landscape analysis. **Information Sciences,** v. 391, p. 28-64, 2017.

JIANG, Q.; JIN, X.; LEE, S. J.; YAO, S. Protein Secondary Structure Prediction: A Survey of the state of the art. **Journal of Molecular Graphics and Modelling,** v. 76, p. 379 – 402, 2017.

JONES, D. T. Genthreader: an efficient and reliable protein fold recognition method for genomic sequences. **Journal of Molecular Biology**, v. 287, n. 4, p. 797 – 815, 1999.

KAUSHIK, R.; SINGH, A.; JAYARAM B. Ab initio Protein Structure Prediction. **Encyclopedia of Bioinformatics and Computational Biology**, v. 1, p. 62-76, 2019.

KALEGARI, D.H.; LOPES, H.S. An improved parallel differential evolution approach for protein structure prediction using both 2d and 3d off-lattice models. **In: 2013 IEEE Symposium on Differential Evolution (SDE)**, Singapura, p. 143–150, 2013.

KATOCH, S.; CHAUHAN, S.S.; KUMAR, V. A review on genetic algorithm: past, present, and future. **Multimedia Tools Applications**, 2020.

KENNEDY, J.; EBERHART, R. Particle swarm optimization. **In: IEEE International Conference on Neural Networks (ICNN95)**, p. 1942 – 1948. Perth, IEEE Press, Australia, 1995.

KITCHENHAM, B.; CHARTERS, S. **Guidelines for performing systematic literature reviews in software engineering**: technical report. 2007.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. **Advances in Neural Information Processing Systems**, v. 25, p. 1097–1105, 2012.

LECUN, Y; BENGIO, Y. Convolutional networks for images, speech, and time series. **The handbook of brain theory and neural networks**, v. 3361, n. 10, p. 1995, 1995

LEGHARI, Z. H.; HASSAN, M. Y.; SAID, D. M.; JUMANI, T. A.; MEMON, Z. A. A novel grid-oriented dynamic weight parameter based improved variant of Jaya algorithm. **Advances in Engineering Software**, v. 150, número do artigo: 102904, 2020.

MAYER, D. G.; KINGHORN, B. P.; ARCHER, A. A. Differential evolution – an easy and efficient evolutionary algorithm for model optimization, **Agricultural Systems**, v. 83, pp. 315–328, 2005.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **The bulletin of mathematical biophysics**, v. 5, n. 4, p. 115 – 133, 1943.

MENG, T.; JING, X.; YAN, Z.; PEDRYCZ, W. A survey on machine learning for data fusion. **Information Fusion**, v. 57, p. 115 – 129, 2020.

MÉSZÁROS, B.; DOBSON, L.; FICHÓ, E.; TUSNÁDY, G. E.; DOSZTÁNYI, Z.; SIMON, I. Sequential, Structural and Functional Properties of Protein Complexes Are

Defined by How Folding and Binding Intertwine. **Journal of Molecular Biology**, v. 431, n. 22, p. 4408 – 4428, 2019.

MISHRA, S.; RAY, P. K. Power quality improvement using photovoltaic fed DSTATCOM based on JAYA optimization. **IEEE Transactions on Sustainable Energy**, v. 7, n. 4, p. 1672 – 1680, 2016.

Mohamed, T. H., Alamin, M. A. M., & Hassan, A. M. Adaptive position control of a cart moved by a DC motor using integral controller tuned by Jaya optimization with Balloon effect. **Computers & Electrical Engineering**, v. 87, número de artigo: 106786, 2020.

MOLINA, D.; POYATOS, J.; SER, J. D.; GARCÍA, S.; HASSAIN, A.; HERRERA, F. Comprehensive Taxonomies of Nature- and Bio-inspired Optimization: Inspiration versus Algorithmic Behavior, Critical Analysis and Recommendations. **Cognitive Computation**, v. 12, p. 897–939, 2020.

MOSS, D.; JELASKA, S.; PONGOR, S. **Essays in Bioinformatics**. [S.l.]: IOS Press, NATO Science Series: Life and Behavioural Sciences, v. 368, 2005.

MOULT, J.; FIDELIS, K.; KRYSHTAFOVYCH, A.; SCHWEDE, T.; TRAMONTANO, A. Critical assessment of methods of protein structure prediction (CASP) - progress and new directions in Round XI. **Proteins: Structure, Function, and Bioinformatics**, v. 84, p. 4–14, 2016.

NELSON, D. L.; COX, M. M. **Princípios de Bioquímica de Lehninger**. 6ª edição. Porto Alegre: Artmed, p. 1328, 2014.

NYGAARD, R.; KIM, J.; MANCIA, F. Cryo-electron microscopy analysis of small membrane proteins. **Current opinion in Structural Biology**, v. 64, p. 26-33, 2020.

OLSON, B.; DE JONG, K.; SHEHU, A. Off-lattice protein structure prediction with homologous crossover. **In: Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation,** Nova York, Estados Unidos da América, p. 287–294, 2013.

OPLS Yale University. Disponível em: http://zarbi.chem.yale.edu/. Acesso em 18 mai. 2021

PARK, S. H.; WU, J.; TAO, Y.; SINGH, C.; TIAN, Y.; MARASSI, F. M.; OPELLA, S. J. Membrane proteins in magnetically aligned phospholipid polymer discs for solid-state NMR spectroscopy. **Biochimica et Biophysica Acta (BBA) – Biomembranes**, v. 1862, n. 9, 2020.

PARPINELLI, R.S.; LOPES, H.S. An ecology-based evolutionary algorithm applied to the 2d-ab off-lattice protein structure prediction problem. **In: 2013 Brazilian Conference on Intelligent Systems,** Fortaleza, Brasil, p. 64–69, 2013.

PEDERSEN, C. N. S. Algorithms in Computational Biology. PhD Thesis, Department of Computer Science. University of Aarhus, Denmark, 2000.

PITZER, E.; AFFENZELLER M. A Comprehensive Survey on Fitness Landscape Analysis. In: Fodor J., Klempous R., Suárez Araujo C.P. (eds) Recent Advances in Intelligent Engineering Systems. **Studies in Computational Intelligence**, v. 378. Springer, Berlin, Heidelberg, Alemanha, 2012.

PLANK, M. **Scientific Autobiography**. 1950.

POE, E. A. **The Power of Words**. Poema publicado em 1845.

PRADHAN, C.; BHENDE, C. N. Online load frequency control in wind integrated power systems using modified Jaya optimization. **Engineering Applications of Artificial Intelligence**, v. 77, p. 212 – 228, 2019.

PRATI, R. C. **Novas abordagens em aprendizado de máquina para a geraçao de regras, classes desbalanceadas e ordenaçao de caso.** Dissertação (Doutorado) - Universidade de São Paulo, p. 191, 2006.

PRICE, K.; STORN, R.M.; LAMPINEN; J.A. **Differential Evolution: A Practical Approach to Global Optimization**. Natural Computing Series. Primeira Edição Springer-Verlag: New York, 2005.

PROCHECK. Disponível em: http://services.mbi.ucla.edu/PROCHECK. Acesso em: 18 mai. 2021.

PROSA. Disponível em: https://www.came.sbg.ac.at/prosa.php. Acesso em: 18 mai. 2021.

QMEAN. Disponível em: https://swissmodel.expasy.org/qmean/. Acesso em: 18 mai. 2021.

RAO, R. V. Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. **International Journal of Industrial Engineering Computations**, v. 7, p. 19-34, 2016.

RAO, R. V.; RAI, D. P., BALIC, J. Optimization of abrasive Waterjet Machining Process using Multi-objective Jaya Algorithm. **Materials Today**, v. 5, p. 4930 – 4938, 2017.

RCSB PDB. **RCSB PDB:** Protein Data Base. 2020. Disponível em: <https://www.rcsb.org/>. Acesso em setembro de 2020.

RIVES, A.; MEIER, J.; SERCU, T.; GOYAL, S.; LIN, Z.; GUO, D.; OTT, M.; ZITNICK, C. L.; MA, J.; FERGUS, R. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequence. **BioArxiv**, pré impressão, 2020.

ROUT, U. K.; SAHU, R. K.; PANDA, S. Design and analysis of differential evolution algorithm based automatic generation control for interconnected power system. **Ain Shams Engineering Journal**, v.4, i. 3, pp. 409-421, 2013.

RUMELHART, D.; HINTON, G.; WILLIAMS, R. Learning representations by back-propagating errors. **Nature**, v. 323, p. 533 – 536, 1986.

RUSSELL, S. J.; NORVIG, Peter. **Artificial intelligence: A modern approach**. Malaysia: Pearson Education Limited, 2016.

SAEED, G. Structural Optimization for Frequency Constraints. **Metaheuristic Applications in Structures and Infrastructures**, pp. 389-417, 2013.

SAR, E.; ACHARYYA, S. Genetic algorithm variants in predicting protein structure. **In: 2014 International Conference on Communication and Signal Processing**, Melmaruvathur, Índia, p. 321–325, 2014.

SCAPIN, M. P. **Um Algoritmo Genético Híbrido Aplicado à Predição da Estrutura de Proteínas Utilizando o Modelo Hidrofóbico-Polar Bidimensional**. Dissertação (Mestrado em Ciências) – Centro Federal de Educação Tecnológica do Paraná. Curitiba, PR, Brasil, p. 153, 2005.

SCHAFFAR, G.; BARRAL, J. M.; BROADLEY, S. A.; HARTL, F. U. Roles of molecular chaperones in protein misfolding diseases. **Seminars in Cell & Developmental Biology**, v. 15, p. 17 – 29, 2004.

SENIOR, A. W.; EVANS, R.; JUMPER, J.; *et al*. Improved protein structure prediction using potentials from deep learning. **Nature**, v. 577, p. 706 – 710, 2020.

SHEN, M.; SALI, A. Statistical potential for assessment and prediction of protein structures. **Protein Science**, v. 15 n. 11, pp. 2507 – 2524, 2006.

SHUKLA, A.; JANMAIJAYA, M.; ABRAHAM, A.; MUHURI, P. K. Engineering applications of artificial intelligence: A bibliometric analysis of 30 years (1988–2018). **Engineering Applications of Artificial Intelligence**, v. 85, p. 517 – 532, 2019.

Silva, R. S.; Parpinelli, R. A Self-adaptive Differential Evolution with Fragment Insertion for the Protein Structure Prediction Problem. **Hybrid Metaheuristics, Springer International Publishing**, p. 136 – 149, 2019.

SINGH, A.; KAUSHIK, R.; MISHRA, A.; SHANKER, A.; JAYARAM, B. ProTSAV: A protein tertiary structure analysis and validation server. **BBA – Proteins and Proteomics**, v. 1864, n. 1, pp. 11 – 19, 2016.

SOHAIB, A. T; QURESHI, S. **An empirical study of Machine Learning techniques for classifying emotional states from EEG data**. Dissertação (Mestrado) — School of Engineering Blekinge Institute of Technology. Karlskrona, Sweden, 2012.

SONG, S.; JI, J.; CHEN, X.; GAO, S.; TANG, Z.; TODO, Y. Adoption of an improved PSO to explore a compound multi-objective energy function in protein structure prediction. **Applied Soft Computing**, v. 72, p. 539-551, 2018.

STILLINGER, F.H.; HEAD-GORDON, T.; HIRSHFEL, C.L. Toy model for protein folding. **Physical Review**, v. 48, p. 1469 – 1477, 1993.

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

SZEGEDY, C.; LIU, W.; JIA, Y.; SERMANET, P.; REED, S.; ANGUELOV, D.; ERHAN, D.; VANHOUCKE, V.; RABINOVICH, A. Going deeper with convolutions. **IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**, p. 1–9, Boston, Estados Unidos da América, 2015.

TAKAYA, D.; NIWA, H.; MIKUNI, J.; NAKAMURA, K.; HANDA, N.; TANAKA, A.; YOKOYAMA, S.; HONMA T. Protein ligand interaction analysis against new CaMKK2 inhibitors by use of X-ray crystallography and the fragment molecular orbital (FMO) method. **Journal of Molecular Graphics and Modelling**, v. 99, 2020.

THIRUMOORTHY, K.; MUNEESWARAN, K. A hybrid approach for text document clustering using Jaya optimization algorithm. **Expert Systems with Applications**, v. 178, número do artigo: 115040, 2021.

TIUMENTSEV, Y.; EGORCHEV, M. **Neural Network Modeling and Identification of Dynamical Systems**. Cambrigde, Massachusetts, Estados Unidos da América: Academic Press, 2019.

TORRISI, M.; POLLASTRI, G.; LE, Q. Deep learning methods in protein structure prediction. **Computational and Structural Biotechnology Journal**, v. 18, p. 1301-1310, 2020.

UNIPROT. **UniProtKB.** 2020. Disponível em: <https://www.uniprot.org/>. Acesso em setembro de 2020.

UNRES. Disponível em: https://unres.pl/. Acesso em 18 mai. 2021

VARGAS, A. C. G.; PAES, A.; VASCONCELOS, C. N. Um estudo sobre redes neurais convolucionais e sua aplicação em detecção de pedestres. **Proceedings of the XXIX Conference on Graphics, Patterns and Images**, p. 1 – 4, 2016.

VENSKE, S. M.; GONÇALVES, R. A.; BENELLI, E. M.; DELGADO, M. R. ADEMO/D: An adaptive differential evolution for protein structure prediction problem. **Expert Systems with Applications**, v. 56, p. 209-226, 2016.

VERIFY 3D. Disponível em: http://services.mbi.ucla.edu/Verify_3D. Acesso em: 18 mai. 2021.

WANG, Y.; MAO, H.; YI, Z. Protein Secondary Structure Prediction by using Deep Learning Method. **Knowledge-Based Systems**, v. 118, p. 115-123, 2016.

WARDAH, W.; KHAN, M. G. M.; SHARMA, A.; RASHID, M. A. Protein secondary structure prediction using neural networks and deep learning: A review. **Computational Biology and Chemistry,** v. 81, p. 1-8, 2019.

WAUDBY, C. A.; DOBSON, C. M.; CHRISTODOULOU J. Nature and Regulation of Protein Folding on the Ribosome. **Trends in Biochemical Science**, v. 44, n. 11, 2019.

XIE, S.; LI, Z.; HU, H. Protein secondary structure prediction based on the fuzzy support vector machine with the hyperplane optimization. **Gene**, v. 648, p. 74-83, 2018.

XU, Y.; XU, D.; LIANG, J. **Computational Methods for Protein Structure Prediction and Modeling:** Volume 1: Basic Characterization. [S.l.]: Springer, 2006.

YADAV, N.; YADAV, A.; KUMAR, M. **An introduction to neural network methods for differential equations**. Netherlands: Springer, 2015.

YAN, K.; WEN, J.; XU, Y.; LIU, B. MLDH-Fold: Protein fold recognition based on multi-view low-rank modeling. **Neurocomputing**, v. 421, p. 127 – 139, 2021.

YON, J. M. Protein folding: a perspective for biology, medicine and biotechnology. **Brazilian Journal of Medical and Biological Research**, v.34, p. 419 – 435, 2001.

YOUSEF, M.; ABDELKADER, T.; EL-BAHNASY, K. Performance comparison of ab initio protein structure prediction methods. **Ain Shams Engineering Journal**, v. 10, p. 713 – 719, 2019.

ZHANG, X.; LI, T. Improved particle swarm optimization algorithm for 2d protein folding prediction. **In: 2007 1st International Conference on Bioinformatics and Biomedical Engineering**, Wuhan, China, p. 53 – 56, 2007.