UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Douglas Monteiro Cavalcanti

Algoritmos de Otimização de Inteligência de Enxame Aplicados ao Problema de Predição de Proteína

Uberlândia, Brasil

2018

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Douglas Monteiro Cavalcanti

Algoritmos de Otimização de Inteligência de Enxame Aplicados ao Problema de Predição de Proteína

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, Minas Gerais, como requisito exigido parcial à obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Profa. Dra. Christiane Regina Soares Brasil

Universidade Federal de Uberlândia – UFU

Faculdade de Computação

Bacharelado em Ciência da Computação

Uberlândia, Brasil

2018

Douglas Monteiro Cavalcanti

Algoritmos de Otimização de Inteligência de Enxame Aplicados ao Problema de Predição de Proteína

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, Minas Gerais, como requisito exigido parcial à obtenção do grau de Bacharel em Ciência da Computação.

Trabalho aprovado. Uberlândia, Brasil, 17 de dezembro de 2018:

Profa. Dra. Christiane Regina Soares Brasil
Orientadora

Prof. Dr. Daniel Antônio Furtado

Profa. Dra. Maria Adriana Vidigal de Lima

Uberlândia, Brasil
2018

Resumo

O problema de predição de estrutura de proteínas é um dos mais desafiadores da área da bioinformática. Muitos dos fatores envolvidos no processo de dobramento ainda não são conhecidos, motivo pelo qual o problema é usualmente tratado a partir de modelos simplificados que consideram apenas algumas poucas características das proteínas, como o modelo Hidrofóbico-Polar 2D. Entretanto, mesmo a partir de modelos simplificados, predizer a estrutura de uma proteína é um problema da classe NP-completo, e por isso, abordagens não determinísticas têm sido largamente aplicadas ao problema. Neste trabalho, dois métodos não determinísticos baseados em Inteligência de Enxame, Otimização por Colônia de Formiga e Otimização por Enxame de Partícula, foram aplicados ao Problema de Estrutura de Proteínas no modelo Hidrofóbico-Polar 2D com inserção do método de busca por *pull move*. Para cada algoritmo três diferentes abordagens foram usadas, cada uma implementando um procedimento diferente de construção de conformações de proteínas. O objetivo do trabalho foi definir qual o melhor método para tratamento do problema, a partir da comparação entre os métodos e suas diferentes abordagens.

Palavras-chave: Otimização por Enxame de Partícula, Otimização por Colônia de Formiga, Predição de Estrutura de Proteínas, Inteligência de Enxame, Modelo 2D HP

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

2D HP	Modelo Hidrofóbico-Polar 2D
3D HP	Modelo Hidrofóbico-Polar 3D
ACO	*Ant Colony Optimization*
EMC	*Evolutionary Monte Carlo*
FR	*Flexible Retrival*
GA	*Genetic Algorithm*
HACO	*Heuristic Ant Colony Optimization*
IA	*Immune Algorithm*
IE	Inteligência de Enxame
PC	*Partial Copy*
PSO	*Particle Swarm Optimization*
PSP	*Protein Structure Prediction*
UH	*Unfold Half*

Lista de símbolos

α	Parâmetro do ACO para influência do feromônio
β	Parâmetro do ACO e PSO para influência da informação heurística
ρ	Parâmetro do ACO para persistência do feromônio
τ_0	Parâmetro do ACO para valor inicial do feromônio
φ_1	Parâmetro do PSO para fator cognitivo.
φ_2	Parâmetro do PSO para fator social.
ω	Parâmetro do PSO para inércia do movimento das partículas

Sumário

1 Introdução

As proteínas são macromoléculas que desempenham funções fundamentais nos organismos vivos, como catálise, transporte, motilidade, armazenamento, regulação e defesa (DOUDNA; COX, 2012). Elas são formadas por uma cadeia linear de aminoácidos. Os aminoácidos são compostos formados por um átomo central de carbono, que está ligado a um agrupamento carboxila ($COOH$), um agrupamento amina (NH_2), um átomo de hidrogênio e um grupo chamado de cadeia lateral, que varia para cada aminoácido. Os aminoácidos que compõem uma proteína são ligados linearmente por meio de ligações peptídicas, formadas a partir da reação do grupo carboxila de um aminoácido com o grupo amina de outro. Em condições adequadas, as proteínas se enovelam (dobram) em uma estrutura tridimensional, assumindo uma conformação estável, apesar das inúmeras possibilidades de dobramento, e alcançando o estado de mais baixa energia do ponto de vista termodinâmico. Esta estrutura é chamada de conformação nativa da proteína, e está diretamente relacionada com a sua função biológica (DOUDNA; COX, 2012).

O conhecimento da conformação de uma proteína oferece informações que auxiliam na compreensão da sua função nos organismos vivos. As possibilidades fornecidas a partir da catalogação da estrutura de uma proteína incluem a realização de testes de hipóteses acerca de sua função, por meio do planejamento da subtituição de aminoácidos em posições definidas da estrutura e, quando consideradas proteínas de relevância médica, o desenho de fármacos que atuem especificamente nas regiões da proteína onde ocorrem as reações biológicas (DOUDNA; COX, 2012). Neste sentido, o estudo e a catalogação das proteínas é de suma importância para as mais diversas áreas da Biologia.

Apesar da importância da catalogação das estruturas tridimensionas das proteínas, apenas uma pequena parcela delas tem sua estrutura conhecida. Isto se deve ao fato de que métodos convencionais para determinação destas estruturas, como difração de raio-x e ressonância nuclear magnética, são limitados, caros e demorados (NELSON; COX, 2015). Neste contexto, diversos estudos na área da Bioinformática vêm sendo realizados com o objetivo de desenvolver algoritmos para predição de estruturas de proteínas a partir de sua sequência linear de aminoácidos, motivados pelo fato de que as instruções para conformação nativa de uma proteína podem estar contidas nesta sequência. Entretanto, ainda não existe um algoritmo definitivo para resolução do problema, pois muitos dos fatores envolvidos no processo de enovelamento são desconhecidos, de modo que ainda não se sabe como as instruções de enovelamento são codificadas na sequência de aminoácidos (DOUDNA; COX, 2012). Acredita-se que o problema de predição exata da estrutura de uma proteína a partir de sua sequência de aminoácidos pode ser resolvido a partir de cálculos computacionais baseados nos conhecimentos empíricos sobre os padrões de eno-

velamento, combinados com a computação teórica de energia, mas atualmente o problema da predição de estrutura de proteínas (PSP, do inglês *Protein Structure Prediction*) continua sem solução, sendo um dos maiores da área da Bioinformática (DOUDNA; COX, 2012).

O processo de enovelamento é extremamente complexo, sendo provocado por diversos critérios intermoleculares e intramoleculares, e podendo ser representado por diversos modelos (DOUDNA; COX, 2012). Por esta razão, a computação do processo pode ser realizada a partir de modelos simplificados, que consideram apenas alguns fatores do processo, fatores estes descobertos a partir da análise empírica de estruturas já conhecidas (DOUDNA; COX, 2012). Estes modelos simplificam tanto a estrutura da proteína quanto a forma como a energia é calculada. Um dos modelos simplificados mais populares para tratamento do problema é o modelo Hidrofóbico-Polar (HP) (LAU; DILL, 1989). Ele se baseia no fato de que os aminoácidos hidrofóbicos na sequência da proteína tendem a se agrupar, auxiliando o dobramento. No modelo, os aminoácidos são representados por pontos em uma malha discreta, que pode ser tanto bidimensional (abordagem chamada de 2D HP) quanto tridimensional (abordagem chamada de 3D HP). As conformações são geradas a fim de enfatizar o contato entre aminoácidos hidrofóbicos. A utilização do modelo simplifica os cálculos computacionais e a busca no espaço de conformações, preservando as reações polares entre os aminoácidos, um dos principais fatores no processo de enovelamento (DOUDNA; COX, 2012; BRASIL, 2012). Mas apesar da simplicidade do modelo, o PSP no modelo HP é um problema de otimização combinatorial da classe NP-completo (CRESCENZI et al., 1998; LAU; DILL, 1989; BRASIL, 2012), e por isso é intratável a partir de algoritmos determinísticos. Neste sentido, os métodos de Inteligência Computacional (IC) são usados como alternativa, por serem capazes de fornecer soluções boas em tempo de execução viável (BRASIL, 2012; KONAR, 2005; SHMYGELSKA; AGUIRRE-HERNANDEZ; HOOS, 2002; BĂUTU; LUCHIAN, 2010).

Estes métodos são estocásticos e baseados em heurísticas, e apesar de não garatirem soluções ótimas, são capazes de gerar soluções boas com um tempo de execução aceitável e têm sido aplicados com sucesso a vários problemas da classe NP-completo (KONAR, 2005). Os algoritmos de Inteligência de Enxame (IE) são exemplos de métodos de IC. Estes algoritmos se baseiam no comportamento de auto-organização de algumas espécies de seres vivos e são voltados principalmente para problemas de otimização, onde o objetivo é encontrar soluções satisfatórias para um problema, que pode ser a melhor dentro de um conjunto de soluções candidatas, ou dentre quaisquer outras soluções (ENGELBRECHT, 2006). Os algoritmos Otimização por Colônia de Formiga (ACO, do inglês *Ant Colony Optimization*) (DORIGO, 1992) e Otimização por Enxame de Partículas (PSO, do inglês *Particle Swarm Optimization*) (EBERHART; KENNEDY, 1995) são um dos principais exemplos de algoritmos de IE, e em anos recentes têm sido amplamente estudados e aplicados ao problema da predição de estrutura de proteína no modelo 2D e 3D HP,

mostrando-se abordagens promissoras para tratamento do problema (YANG et al., 2018; LLANES et al., 2016; THILAGAVATHI; AMUDHA, 2015; XIAO; LI; HU, 2014; LIU; YANG, 2014; CITROLO; MAURI, 2013; MANSOUR; KANJ; KHACHFE, 2012; JANA; SIL, 2012; LIN; SU, 2011; BĂUTU; LUCHIAN, 2010; FIDANOVA; LIRKOV, 2008; HU; ZHANG; LI, 2008; SHMYGELSKA; AGUIRRE-HERNANDEZ; HOOS, 2002). Métodos auxiliares de busca têm sido incorporados aos algoritmos, a fim de melhorar a qualidade das soluções geradas, podendo destacar-se o método de busca por *pull move*, bem como métodos para correção de conformações inviáveis geradas em tempo de execução, dada a característica restritiva do PSP no modelo HP, onde conformações que levam à colisão de aminoácidos definem regiões inviáveis do espaço de busca, que devem ser evitadas pelos algoritmos.

1.1 Objetivos

O objetivo geral deste trabalho foi comparar a eficácia dos algoritmos ACO e PSO na predição de estrutura de proteínas no modelo 2D HP, ambos com o método de busca por *pull move*, a partir de três diferentes abordagens.

Os objetivos específicos foram:

- Comparar diferentes métodos de construção de conformações de proteínas, a fim de escolher aquele que melhor otimiza os resultados alcançados pelos algoritmos;
- Comparar a melhor versão encontrada para o ACO e o para o PSO entre si, a fim de definir qual o algoritmo de melhor desempenho para o PSP no modelo 2D HP.
- Comparar a melhor versão encontrada para o ACO e o PSO com resultados da literatura, a fim de validá-los como propostas viáveis para tratamento do PSP no modelo 2D HP.

1.2 Motivação

O estudo e o desenvolvimento de métodos capazes de tratar com sucesso o PSP no modelo HP caracterizam uma contribuição para o entendimento, ainda que de maneira simplificada, do modo como as reações de hidrofobicidade dos aminoácidos influenciam no enovelamento. Esta contribuição qualifica um avanço na compreensão de como a sequência de aminoácidos de uma proteína codifica as instruções de enovelamento e na resolução do PSP de modo geral (DOUDNA; COX, 2012; LAU; DILL, 1989; BRASIL, 2012). A comparação entre estes métodos permite a análise de suas vantagens e desvantagens, ajudando na elaboração de métodos híbridos e eficazes para o problema em trabalhos futuros.

1.3 Organização do Trabalho

Este trabalho está organizado da seguinte forma:

- O Capítulo 2 detalha o problema PSP e o Modelo 2D HP.
- O Capítulo 3 descreve os métodos ACO e PSO, e apresenta um histórico de trabalhos relacionados, que aplicaram os algoritmos ao problema PSP no Modelo HP;
- O Capítulo 4 apresenta os algoritmos implementados;
- O Capítulo 5 apresenta, analisa e compara os resultados alcançados para ambos os algoritmos;
- O Capítulo 6 apresenta as conclusões, bem como as contribuições deste trabalho.

2 Problema de Predição de Estrutura de Proteínas

O Problema de Predição de Estrutura de Proteínas é um dos mais desafiadores da área da bioinformática e consiste em tentar predizer a estrutura tridimensional que uma proteína adota após o processo de dobramento. Muitos dos fatores envolvidos neste processo são ainda desconhecidos, motivo pelo qual o problema é normalmente tratado a partir de modelos simplificados que consideram apenas algumas características das proteínas. Mas mesmo a partir de modelos simplificados, predizer a estrutura de uma proteína ainda é um problema da classe NP-completo, não havendo, portanto, abordagem determinística capaz de tratá-lo.

2.1 Estrutura da Proteínas

As proteínas são formadas por uma sequência linear de aminoácidos ligados de modo covalente [1]. Cada proteína possui uma sequência de aminoácidos única que confere a ela uma determinada estrutura tridimensional. Esta estrutura, por sua vez, confere uma função específica para proteína. Deste modo, proteínas com funções diferentes sempre possuem sequências de aminoácidos diferentes (NELSON; COX, 2015).

A estrutura de uma proteína é definida em quatro níveis hierárquicos. A sequência linear de aminoácidos da proteína descreve sua estrutura primária; enquanto que o arranjo espacial de resíduos de aminoácidos adjacentes em um segmento da proteína define uma estrutura secundária. A estrutura terciária é descrita pelo arranjo tridimensional total de todos os átomos da proteína; quando uma proteína apresenta mais de uma subunidade de estrutura terciária, a disposição espacial destas subunidades define sua estrutura quaternária. (NELSON; COX, 2015).

2.1.1 Estrutura Primária

Os aminoácidos são os blocos de construção das proteínas. Uma aminoácido é formado por um grupamento carboxila COOH, um agrupamento amina NH_3 e uma cadeia lateral R, ligados a um mesmo átomo de carbono, chamado de carbono alpha, denotado por C_α. A cadeia lateral R distingue um aminoácido dos demais e define suas propriedades químicas, como a polaridade. Dois aminoácidos podem ser ligados de modo covalente por meio da desidratação (remoção dos elementos de água) do grupamento amino de um

[1] Ligação covalente é uma ligação que ocorre a partir do compartilhamente de um ou mais pares de elétrons entre átomos, não havendo transferência de elétrons.

aminoácido e do grupamento carboxila de outro (ver Figura 1). A ligação covalente entre dois aminoácidos adjacentes é chamada de ligação peptídica, e o produto de diversas ligações deste tipo é conhecido como cadeia peptídica. No processo de desidratação, o aminoácido perde alguns de seus átomos, e por isso, a partir do momento que passa a fazer parte de uma cadeia peptídica, é usualmente chamado de resíduo de aminoácido. As proteínas são cadeias peptídicas formadas a partir de um conjunto de aminoácidos, combinados em diferentes sequências. A sequência de resíduos de aminoácidos de uma proteína descreve a sua estrutura primária (NELSON; COX, 2015; DOUDNA; COX, 2012).

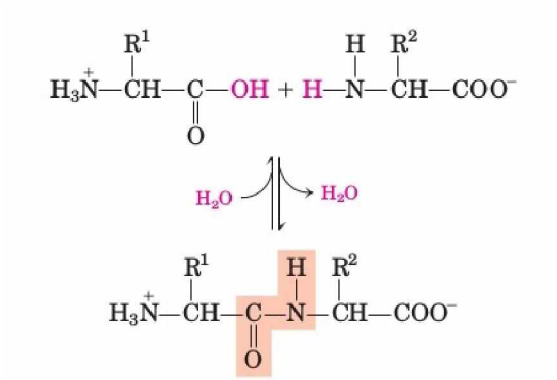

Figura 1 – Dois aminoácidos reagem formando uma ligação peptídica. Adaptado de (NELSON; COX, 2015).

2.1.2 Estrutura Secundária

As ligações peptídicas são planares, isto é, os átomos que compõe a ligação estão em um mesmo plano, e por isso não há rotação livre ao redor delas. Porém, existe flexibilidade ao redor da ligação N–C_α e da ligação C_α–C. Desta forma, o esqueleto de uma cadeia polipeptídica pode ser descrito como uma série de planos rígidos, onde planos consecutivos compartilham um ponto de rotação em comum (ver Figura 2). Os ângulos onde é possível a rotação são chamados de ângulos diedrais e permitem que os átomos da cadeia principal adotem diferentes conformações, estabilizadas por pontes de hidrogênio, formando padrões repetitivos e regulares em determinados segmentos da proteína. Estas conformações descrevem as estruturas secundárias e ocorrem extensamente nas proteínas. As principais estruturas secundárias são as hélices α, conformações β e voltas β. Quando um padrão regular não é observado, a estrutura secundária é chamada de indefinida ou espiral aleatória, não descrevendo adequadamente a estrutura do segmento.

Hélice α: Nesta estrutura, o esqueleto polipeptídico é enrolado em torno de um eixo imaginário, formando uma espiral (ver Figura 3(a)). Cada volta da hélice contém cerca de 3,6 resíduos de aminoácidos, sendo que os grupos R dos resíduos são projetados para fora do esqueleto helicoidal. Apesar de ser uma estrutura comum, nem todos

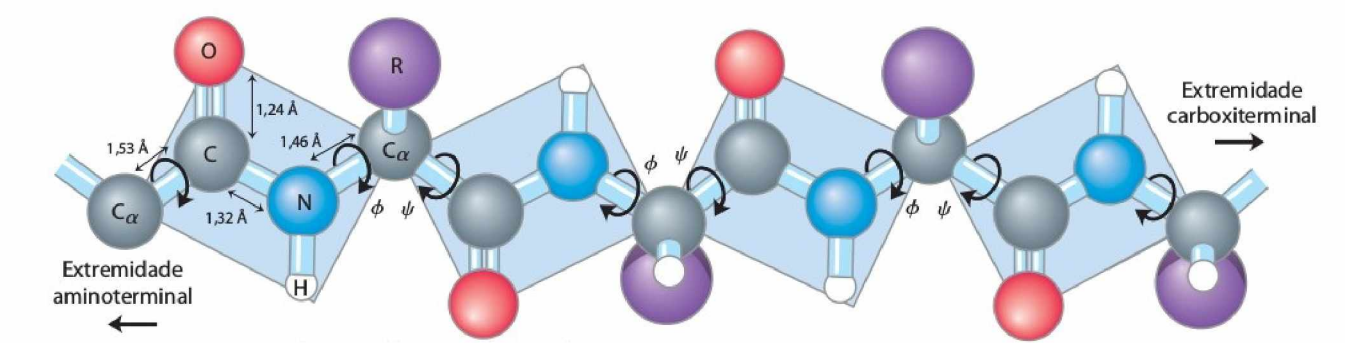

Figura 2 – Esqueleto de uma cadeia peptídica. ψ e ϕ representam ângulos diedrais. Adaptado de (NELSON; COX, 2015).

os polipetídeos podem formar uma hélice α estável. As propriedades dos grupos R e a capacidade dos átomos de aceitar os ângulos diedrais característicos definem a propensão do resíduo de uma polipeptídeo de formar uma hélice α. A posição do resíduo em relação a seus vizinhos também é um fator importante.

Conformação β: Nesta estrutura, o esqueleto polipeptídico é estendido em forma de zigue-zague. O arranjo de vários segmentos na conformação β lado a lado é chamado de **folha** β. A estrutura de uma folha β é estabilizada pelas ligações de hidrogênio formadas entre as conformações β (ver Figura 3(b)) dispostas de maneira paralela ou antiparalela. Aminoácidos não favoráveis à formação da estrutura hélice α podem ser observados com frequência nas folhas β.

Volta β: Nesta estrutura, o esqueleto polipeptídico forma uma volta de 180° envolvendo quatro resíduos de aminoácidos, que conecta as extremidades de duas conformações β adjacentes de uma folha β antiparalela. A estrutura é estabilizada a partir de uma ligação de hidrogênio do átomo de oxigênio do primeiro resíduo com o hidrogênio do grupo amina do quarto resíduo (ver Figura 3(c)).

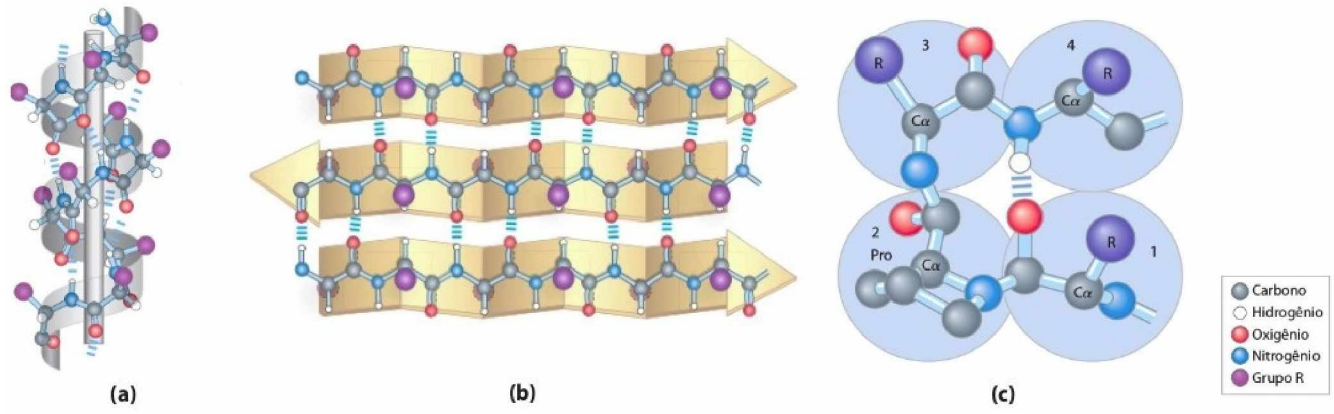

Figura 3 – (a) Estrutura Hélice α. (b) Estrutura Folha β. (c) Estrutura Volta β. Adaptado de (NELSON; COX, 2015).

2.1.3 Estrutura Terciária e Quaternária

A estrutura terciária de uma proteína é definida pela orientação tridimensional dos diferentes elementos de estrutura secundária. Aminoácidos distantes na cadeia polipeptídica podem interagir uns com os outros na estrutura da proteína completamente dobrada. A conformação e a estabilidade dessa estrutura são influenciadas por vários fatores, incluindo interações de van der Waals, força eletrostática, pontes de hidrogênio entre os resíduos ou com solvente, e o efeito hidrofóbico. Uma proteína pode conter duas ou mais cadeias polipeptídicas distintas. A disposição das subunidades de uma proteína em complexos tridimensionais descreve a sua estrutura quaternária (ver Figura 4) (NELSON; COX, 2015; DOUDNA; COX, 2012).

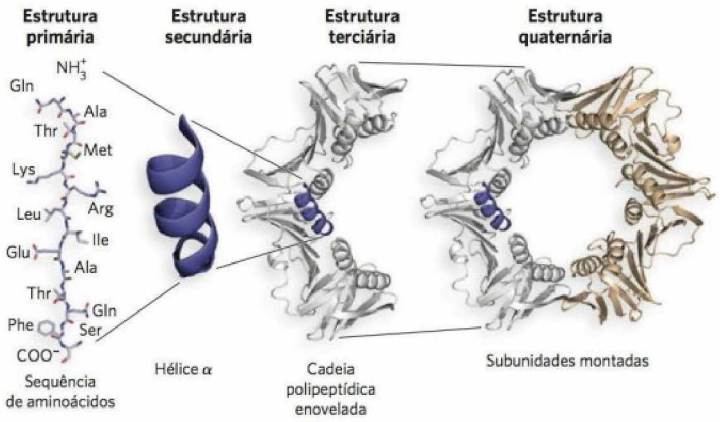

Figura 4 – Os quatro níveis estruturais de uma proteína. Adaptado de (DOUDNA; COX, 2012).

Apesar das várias possibilidades de conformações que uma proteína pode adotar, poucas predominam em condições biológicas. A estrutura tridimensional que uma proteína adota é aquela termodinamicamente mais estáveis. Esta estrutura é chamada de conformação funcional, pois é ela quem define a função biológica da proteína. Quando uma proteína se encontra enovelada em uma conformação funcional, é chamada de proteína nativa (NELSON; COX, 2015; DOUDNA; COX, 2012).

2.2 O Enovelamento Protéico

O processo pelo qual a cadeia polipeptídica se dobra para alcançar o estado de mais baixa energia do ponto de vista termonidinâmico é chamado de enovelamento proteíco (DOUDNA; COX, 2012).

Até 1968 acreditava-se que as proteínas se enovelavam de maneira aleatória, mediante tentativas de todas conformações possíveis para sua cadeia de aminoácidos. Foi Cyrus Levinthal quem postulou a impossibilidade do processo aleatório de enovelamento (DOUDNA; COX, 2012). Supondo uma proteína composta por uma cadeia de 100 resíduos

de aminoácidos e assumindo que cada resíduo pode adotar 10 conformações diferentes, o espaço conformacional do polipeptídeo é de 10^{100} conformações. Se esta proteína se enovelasse por meio do processo aleatório de enovelamento, atingindo cada conformação no menor tempo possível, 10^{-13} segundos, que é o tempo necessário para uma vibração molecular (DOUDNA; COX, 2012), seriam necessários 10^{77} anos para que a proteína experimentasse todas as possíveis conformações. Em constrate a este fato, existem proteínas de 100 aminoácidos biologicamente ativas produzidas pelos organismos vivos em cerca de 5 segundos. Esta contradição ficou conhecida como Paradoxo de Levinthal, e mostra que o enovelamento não pode ser aleatório, mas um processo ordenado que evita a maior parte das conformações intermediárias possíveis (DOUDNA; COX, 2012).

A estrutura primária de uma proteína contém as instruções necessárias para seu enovelamento, entretanto, os pesquisadores ainda não são capazes de prever com exatidão a estrutura tridimensional da proteína apenas pela sua sequência de aminoácidos. Apesar de se conhecer as principais forças que controlam o enovelamento das proteínas, a energia de uma proteína no estado enovelado e não enovelado é muito pequena, o que dificulta o entedimento do código de enovelamento (DOUDNA; COX, 2012).

A medida que uma proteína se enovela, o espaço de busca conformacional fica cada vez mais restrito, e a energia livre da macromolécula é reduzida. Na Figura 5 são mostrados diferentes exemplos de curvas tridimensionais representando o espaço conformacional, onde quanto mais profundo no funil formado, menor a energia livre da proteína. O ponto de menor energia livre na curva representa a conformação nativa da proteína. Outras depressões na superfície representam mínimos locais, intermediários de enovelamento com estabilidade significativa, mas que não dizem respeito a estrutura funcional da proteína. O processo de enovelamento consegue minimizar a energia livre da proteína, evitando todos os estados intermediários e alcançando a conformação nativa de maneira eficiente (NELSON; COX, 2015).

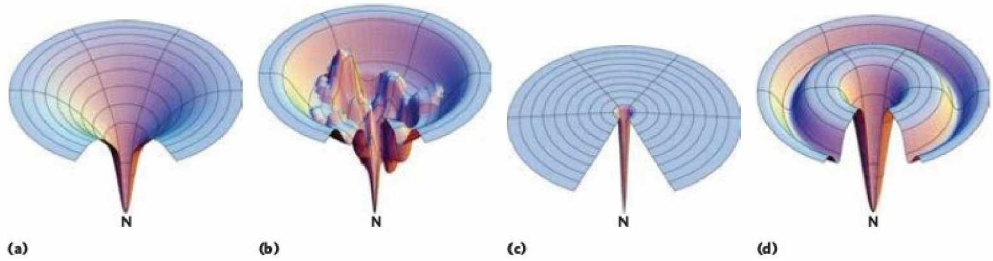

Figura 5 – Representação de espaços conformacionais de proteínas. Adaptado de (NELSON; COX, 2015).

2.3 Problema de Predição de Estrutura de Proteínas

O PSP consiste em tentar prever a estrutura terciária de uma proteína a partir de sua estrutura primária. Na modelagem computacional de métodos para tratamento deste problema, duas abordagens têm sido estudadas: a modelagem baseada em conhecimento e a modelagem *ab initio* (BRASIL, 2012).

2.3.1 Modelagem baseada em conhecimento

Nesta abordagem, são desenvolvidas técnicas que tentam predizer a estrutura de uma proteína a partir de uma proteína similar cuja estrutura já se conhece. Deste modo, técnicas baseadas nesta abordagem são limitadas pela quantidade de moléculas já conhecidas, pois é necessário que a proteína cuja estrutura deseja-se conhecer apresente semelhanças com as proteínas de estruturas já catalogadas. Dentre as técnicas que se baseiam nesta modelagem, destacam-se a modelagem por homologia e por *threading* (BRASIL, 2012).

Modelagem baseada em homologia: Na modelagem por homologia, procura-se prever a estrutura da proteína alvo, utilizando a estrutura de uma proteína de sequência homóloga, chamada de proteína molde, alinhando a proteína alvo à proteína molde por meio de modelos computacionais.

Modelagem baseada em *threading*: Na modelagem por *threading*, o espaço de busca conformacional é reduzido a um conjunto de dobramentos de estruturas já catalogadas, as quais não possuem necessariamente sequência homóloga à sequencia da proteína alvo. Desta forma, o dobramento da proteína de interesse é buscado diretamente nas estruturas tridimensionais conhecidas. Essa abordagem se baseia no fato de que muitas das proteínas de estrutura catalogada possuem estruturas terciárias similares, mesmo com estruturas primárias relativamente menos similares.

2.3.2 Modelagem *ab initio*

Nesta abordagem a predição é feita conhecendo-se unicamente a estrutura primária da molécula e por isso não é necessária a comparação com moléculas de estruturas já conhecidas. Para isso, realiza-se uma busca no espaço conformacional a fim de encontrar a estrutura de menor energia livre (BRASIL, 2012; NELSON; COX, 2015).

Há, entretanto, uma dificuldade em construir modelos computacionais confiáveis para a tarefa, pois ainda existem muitas questões não resolvidas sobre os fatores responsáveis pelo enovelamento proteico. Devido a esta limitação, são desenvolvidos modelos simplificados que abstraem algumas questões do processo de enovelamento, como o cálculo da energia livre ou o próprio espaço de busca, a um nível de detalhe desejado (BRASIL,

2012; NELSON; COX, 2015).

2.3.3 Modelos de representação das proteínas

Com o objetivo de diminuir a complexidade no tratamento do PSP vários modelos foram propostos. Estes modelos simplificam tanto a estrutura da proteína quanto a forma como a energia é calculada e podem ser divididos em modelos baseados em *lattice*, *off-lattice* e *full-atom* (DILL et al., 1995; BRASIL, 2012).

Modelo *lattice*: Neste modelo, cada aminoácido da proteína é representado como um ponto. As ligações entre os aminoácidos são representadas por linhas. Os aminoácidos são distribuídos em uma malha dividida em unidades que podem conter até um aminoácido cada. Os ângulos das ligações podem assumir apenas alguns poucos valores discretos, definidos pela estrutura da malha. Diversos tipos de malhas são possíveis, sejam elas bidimensionais ou tridimensionais. Algumas características dos aminoácidos podem ser usadas para avaliar a conformação da proteína neste modelo, como as interações polares. O Modelo HP é um exemplo de modelo lattice que utiliza as interações polares para avaliar a conformação gerada.

Modelo *off-lattice*: Neste modelo os ângulos diedrais podem assumir valores contínuos dentro de intervalos específicos, permitindo a simulação de interações energéticas mais realistas. A proteína pode mover-se livremente no espaço de busca, não sendo, portanto, restringida a uma rede. O modelo também permite que os átomos da cadeia de aminoácidos sejam representados na simulação, possibilitando uma representação mais realista da proteína.

Modelo *full-atom*: Este modelo representa a estrutura de uma proteína a partir de seus ângulos diedrais, representados por um conjunto de quatro átomos conectados. Estes ângulos podem girar livremente, e pequenas variações em seus valores podem mudar significativamente a estrutura da proteína.

2.4 O Modelo Hidrofóbico-Polar

O Modelo HP (LAU; DILL, 1989) é baseado em *lattice* e utiliza a propriedade da hidrofobicidade [2] dos aminoácidos da proteína. Neste modelo, são diferenciados aminoácidos hidrofóbicos (H) dos aminoácidos hidrofílicos, ou polares (P). As conformações são geradas a fim de enfatizar o contato entre aminoácidos hidrofóbicos. No modelo, os aminoácidos subsequentes na estrutura primária ocupam posições adjacentes na rede e são ditos conectados. Os aminoácidos não conectados que ocupam posições adjacentes na malha são chamados de vizinhos. As posições adjacentes distanciam em uma unidade

[2] Hidrofobicidade é a capacidade da molécula de repelir água.

para qualquer uma das configurações. A energia mínima de uma conformação é dada pelo número negativo de adjacências entre aminoácidos H. A representação da conformação da proteína pode ser feita tanto em uma malha bidimensional (abordagem chamada de 2D HP) quanto em uma malha tridimensional (abordagem chamada de 3D HP).

Seja uma proteína no Modelo HP dada pela sequência S definida como $s_1 s_2 \dots s_n$, onde s_i denota o i-ésimo aminoácido da sequência e n o comprimento da proteína. A energia mínima entre o i-ésimo aminoácido e o j-ésimo aminoácido é dada pela Equação 2.1

$$E_{ij} = \begin{cases} -1, & \text{se } s_i \text{ e } s_j \text{ são ambos do tipo H} \\ 0, & \text{caso contrário} \end{cases} \tag{2.1}$$

A energia mínima em uma proteína é dada pelas Equações 2.2 e 2.3:

$$E = \Delta r_{ij} E_{ij} \tag{2.2}$$

$$\Delta r_{ij} = \begin{cases} 1, & \text{se } s_i \text{ e } s_j \text{ são vizinhos} \\ 0, & \text{caso contrário} \end{cases} \tag{2.3}$$

3 Algoritmos de Inteligência de Enxame

Os fenômenos de auto-organização encontrados na natureza têm servido de inspiração para o desenvolvimento de métodos computacionais de otimização capazes de tratar problemas para os quais não se conhecem abordagens analíticas. As técnicas ACO e PSO são os principais métodos baseados nestes fenômenos. Estas técnicas são classificadas como algoritmos de Inteligência de Enxame e têm seu funcionamento baseado em regras simples e informações locais, e não necessitam de um coordenador central, além de serem robustos ao desvio de alguns indivíduos. Neste capítulo ambas as técnicas serão descritas com mais detalhes.

3.1 Otimização por Colônia de Formiga (ACO)

A técnica ACO é uma metaheurística originalmente introduzida em (DORIGO, 1992), para problemas de otimização combinatorial, inspirada no comportamento das formigas ao trilhar um caminho entre o formigueiro e a fonte de alimento. O primeiro algoritmo baseado na metaheurística foi desenvolvido para aplicação no Problema do Caixeiro Viajante (DORIGO; MANIEZZO; COLORNI, 1996). Desde então algoritmos baseados nesta metaheurística foram aplicados com sucesso a vários problemas de otimização combinatorial (YAN; SHI, 2011; RAN; LIU; YANG, 2013; LLANES et al., 2016).

3.1.1 Funcionamento

O funcionamento do ACO se baseia na construção de soluções candidatas para um problema de otimização combinatorial. Estas soluções são construídas por formigas artificiais que decidem como compôr suas soluções baseadas em uma regra probabilística que considera eventuais informações heurísticas sobre os componentes a serem combinados e as informações de uma memória compartilhada, baseada em trilhas de feromônio[1]. Ao concluir o procedimento de contrução, cada formiga deposita uma quantidade de feromônio nos componentes da solução construída. Deste modo, componentes mais usados ganham destaque entre os outros, aumentando a probabilidade de que sejam escolhidos novamente. O processo é repetido até que um critério de parada seja atingido (DORIGO, 1992; DORIGO; MANIEZZO; COLORNI, 1996).

Um dos elementos fundamentais do algoritmo é o processo de evaporação do feromônio. Este processo garante que trilhas geradas em momentos mais avançados do algoritmo consigam competir com trilhas mais antigas, já consolidadas pelo feromônio. Por

[1] Feromônio é uma substância química utilizada para comunicação entre indíviduos de uma mesma espécie.

meio da evaporação, o feromônio acumulado nos componentes é reduzido a cada iteração. Desta maneira, trilhas cuja qualidade já tenha sido superada, são esquecidas conforme é diminuída a frequência de formigas que optam por elas, contrabalanceando a avaliação positiva dos componentes.

A seguir, é apresentado o procedimento computacional referente ao ACO:

Algoritmo 1: ACO

```
1: inicialize as trilhas de feromônio
2: enquanto critério de parada não for atingido faça
3:    para cada formiga faça
4:       construa solução candidata
5:    fim para
6:    atualize as trilhas de feromônio
7: fim enquanto
```

Em Dorigo, Maniezzo e Colorni (1996), os autores adaptaram o algoritmo ACO proposto, incorporando à técnica um método auxiliar de busca local, aplicado às soluções construídas pelas formigas a fim de buscar modificações locais capazes de melhorar a qualidade das soluções. Os resultados obtidos mostraram que o uso do método de busca local é capaz de aumentar significativamente o desempenho do ACO.

3.1.2 Regras e Parâmetros

A regra probabilística para construção das soluções é dada pela Equação 3.1:

$$p_{ij} = \frac{(\tau_{ij})^{\alpha}(\eta_{ij})^{\beta}}{\sum_{j=1}^{n}(\tau_{ij})^{\alpha}(\eta_{ij})^{\beta}} \tag{3.1}$$

onde p_{ij} é a probabilidade de uma formiga incluir o componente j na solução, τ_{ij} é a intensidade do feromônio na aresta que conecta o componente i ao componente j, η_{ij} é a informação heurística associada a esta mesma aresta, n é o número de componentes disponíveis, α é um valor real positivo que regula a influência do feromônio e β é um valor real positivo que regula a influência da informação heurística.

A evaporação do feromônio é realizada seguindo a Equação 3.2:

$$\tau_{ij} = \rho\tau_{ij} \tag{3.2}$$

onde ρ é um valor real no intervalo de 0 a 1, que controla a taxa de persistência do feromônio.

O depósito de ferômonio é executado por cada formiga seguindo a Equação 3.3:

$$\tau_{ij} = \tau_{ij} + \frac{1}{l} \tag{3.3}$$

onde l é o custo da solução.

No início do algoritmo os componentes são inicializados com um valor arbitrário de feromônio, denotado por τ_0.

3.1.3 Trabalhos Relacionados

Shmygelska, Aguirre-Hernandez e Hoos (2002) introduziram um algoritmo ACO para o PSP 2D HP. Fidanova e Lirkov (2008) propuseram um algoritmo ACO para o PSP 3D HP. Hu, Zhang e Li (2008) propuseram um novo algoritmo ACO para o PSP 2D HP, chamado de *Flexible Ant Colony Algorithm* (FAC), que utiliza um método de *backtracking* para reparar conformações inválidas geradas em tempo de execução. Ran, Liu e Yang (2013) propuseram um algoritmo híbrido entre o ACO e Markov Chain Monte Carlo para o PSP HP. Liu e Yang (2014) desenvolveram uma versão do ACO para o PSP HP usando o método de busca por *pull moves* como técnica de busca local e o método de cópia parcial para gerenciar conformações inviáveis, a nova abordagem foi chamada de *Heuristic Ant Colony Optimization* (HACO). Thilagavathi e Amudha (2015) aplicaram ao PSP HP uma versão do ACO baseada em classificação, chamada de *Rank Based Ant Colony Optimization*. Llanes et al. (2016) propuseram uma versão paralela do ACO aplicada ao PSP HP, usando *Compute Unified Device Architecture* (CUDA). Wang (2018) propôs um método híbrido do ACO com *Artificial Fish Swarm Algorithm* [2] para o PSP 2D HP.

3.2 Otimização por Enxame de Partículas

Originalmente introduzido em (EBERHART; KENNEDY, 1995), o PSO é uma meta-heurística para otimização de funções contínuas não lineares, fundamentada a partir do desenvolvimento de simulações simplificadas da coreografia imprevisível de uma revoada de pássaros.

3.2.1 Funcionamento

Nesta técnica de otimização, soluções candidatas para o problema são representadas por conjunto de partículas distribuídas no espaço de busca. A coordenada de uma partícula define a solução representada por ela. Cada partícula é avaliada quanto a qualidade de sua posição no espaço de busca por meio da função definida pelo problema alvo, chamada de função *fitness*. Cada partícula possui uma taxa de variação de sua posição, chamada de velocidade, e mantém um registro de sua melhor posição prévia, chamada de *pbest*. A melhor posição dentre todas as posições encontradas pela população de partículas é chamada de *gbest* e é conhecida por todo enxame (EBERHART; KENNEDY, 1995).

[2] Algoritmo inspirado no movimento coletivo dos peixes (NESHAT et al., 2014)

O algoritmo é iniciado com a população de partículas distribuída aleatoriamente sobre o espaço de busca, e a cada iteração, para cada partícula são calculados os vetores diferença entre a posição atual da partícula e as posições *pbest* e *gbest*. A velocidade da partícula é ajustada na direção da soma destes vetores, podendo ser limitada a um valor máximo.

A seguir, é apresentado o procedimento computacional referente ao PSO:

Algoritmo 2: PSO

```
1: inicialize a posição e a velocidade das partículas
2: para cada iteração faça
3:    para cada partícula faça
4:       atualize a velocidade da partícula
5:       atualize a posição da partícula
6:    fim para
7: fim para
```

3.2.2 Regras e Parâmetros

A velocidade v de uma partícula e sua posição x na iteração $i+1$ são calculadas seguindo as Equações 3.4 e 3.5 (SHI; EBERHART, 1998):

$$v_{i+1} = v_i w + r_1 \varphi_1 (pbest - x_i) + r_2 \varphi_2 (gbest - x_i) \tag{3.4}$$

$$x_{i+1} = x_i + v_{i+1} \tag{3.5}$$

A influência da posição *pbest* sobre o cálculo da velocidade é controlada pelo fator cognitivo, denotado por φ_1, enquanto que a influência da posição *gbest* é controlada pelo fator social, denotado por φ_2. Estes parâmetros podem assumir qualquer valor real positivo. Os fatores r_1 e r_2 são dois valores randômicos dentro do intervalor de 0 a 1, definidos a cada iteração, que multiplicam os fatores cognitivo e social, respectivamente, inserindo um fator aleatório na movimentação das partículas. O controle da influência da velocidade v_i sobre a velocidade v_{i+1} é feita pelo parâmetro w, chamado de peso de inércia, cujo valor pode variar no intervalo de 0 a 1.

3.2.3 Trabalhos Relacionados

Băutu e Luchian (2010) projetaram um variante do PSO discreto, chamado de *Roulette* PSO (RPSO). O algoritmo foi inspirado no PSO Binário e no método da roleta dos Algoritmos Genéticos (AG), e desenvolvido para aplicação em modelos de enovelamento de proteínas baseados em redes. Lin e Su (2011) propuseram um algoritmo híbrido

entre PSO e AG para aplicação no PSP 3D HP. Em 2012, dois novos algoritmos baseados no PSO para aplicação no PSP HP foram propostos. O primeiro deles, de Mansour, Kanj e Khachfe (2012), foi desenvolvido para aplicação no mesmo PSP 3D HP, o segundo, de Jana e Sil (2012), para o PSP 2D HP, utilizando um método de *backtracking* para reparar conformações inválidas geradas em tempo de execução. Xiao, Li e Hu (2014) desenvolveram uma nova abordagem do PSO baseada no algoritmo *set-based* PSO (CHEN et al., 2010), para aplicação nos modelos em rede 2D e 3D HP. Yang et al. (2018) desenvolveram uma versão do PSO chamada de *Hybrid High Exploration Particle Swarm Optimization*, que combina um algoritmo guloso e a Busca de Subida da Encosta como algoritmos de busca local.

4 Desenvolvimento

Neste capítulo serão detalhados os algoritmos implementados para o ACO e PSO, bem como o método de busca por *Pull Move* (LESH; MITZENMACHER; WHITESIDES, 2003), incorporado a ambos os algoritmos. Serão detalhadas também as três diferentes versões implementadas para o ACO e para o PSO.

4.1 Algoritmo ACO para o PSP no Modelo HP

No algoritmo implementado para este trabalho, cada formiga constrói uma conformação posicionando um a um os aminoácidos na malha. O procedimento de construção consiste em uma laço que itera sobre todas as arestas da proteína, definindo uma direção de enovelamento para cada uma delas.

As conformações são representadas por uma sequência de direções relativas de enovelamento, que indicam a posição de um aminoácido com relação ao seu antecessor. As possíveis direções que uma aresta pode adotar são: direita (R, do inglês *right*), esquerda (L, do inglês *left*) e reto (S, do *inglês straight*). A escolha sobre qual direção uma aresta irá assumir é feita usando uma regra probabilística que leva em conta o número de novos contatos hidrofóbicos que a dobra irá gerar, e um valor de feromônio associado a cada possível direção. Por exemplo, a partir do segundo aminoácido H de uma sequência hipotética, a formiga tinha três possíveis movimentos na malha (Figura 6 (a)). Foi escolhido o movimento para direita, e colocou-se o aminoácido P, tendo esta posição três novas possibilidades de movimentação, relativas ao último movimento realizado (Figura 6 (b)).

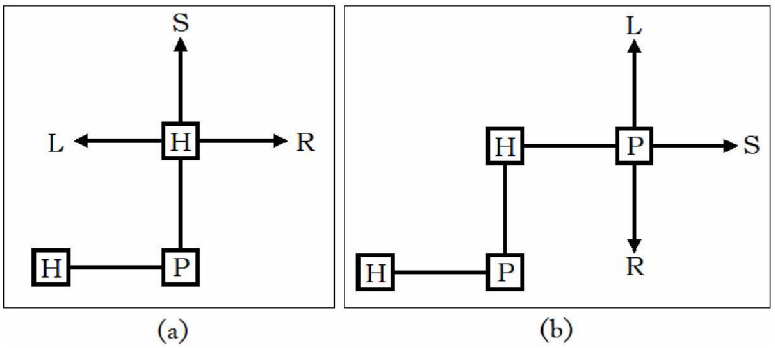

Figura 6 – Possíveis direções que uma aresta pode adotar.

A estrutura principal do algoritmo é definida por 3 laços, o laço mais interno, que itera sobre os aminoácidos, o laço intermediário, que itera sobre as formigas da população, e o laço mais externo, que é executado até que um número máximo de iterações seja atingido.

O algoritmo é divido em 3 fases: A fase de construção, a fase de busca local e a fase de atualização do feromônio. Essas fases são executadas em sequências pelo algoritmo. Ao final do procedimento, a conformação de menor energia encontrada é retornada como solução. Em seguida é detalhada cada umas das fases do algoritmo.

4.1.1 Fase de Construção

Três abordagens distintas são comumente usadas para a etapa de construção da conformação. A primeira delas inicia a construção posicionando os dois primeiros aminoácidos da sequência na malha, e define a direção de cada aresta a partir da primeira, seguindo a ordem crescente da sequência de arestas. Na segunda abordagem, a conformação é construída a partir do meio da sequência de aminoácidos. Neste caso, no início da etapa de construção da conformação de uma proteína com N aminoácidos, os aminoácidos de índices $\lfloor \frac{N}{2} \rfloor$ e $\lfloor \frac{N}{2} \rfloor + 1$ são posicionados lado a lado no centro da malha. Alternando entre a extremidade da esquerda e a extremidade da direita, a conformação é construída até que sejam posicionados os aminoácidos de índice 0 e $N - 1$. Na terceira abordagem, a conformação é construída a partir de uma aresta escolhida aleatoriamente de maneira independente para cada formiga, com os aminoácidos respectivos a estas arestas sendo posicionados lado a lado na malha. O processo de construção é, deste modo, executado de maneira análoga a segunda abordagem.

Ainda no contexto da construção de uma conformação, uma conformação é dita válida se evita a sobreposição de aminoácidos. Neste sentido, entende-se como colisão o estado em que o algoritmo de construção não consegue prosseguir sem que haja sobreposição de aminoácidos. Conformações no estado de colisão são chamadas de conformações inviáveis. O gerenciamento destas conformações é realizado por meio de mecanismos de correção, geralmente atrelados ao procedimento de construção.

Os procedimentos de construção implementados foram baseados em (SHMYGELSKA; AGUIRRE-HERNANDEZ; HOOS, 2002), que desfaz metade da conformação construída quando ocorre uma colisão, (HU; ZHANG; LI, 2008), que divide a conformação em duas subconformações e aplica o seu mecanismo de correção em apenas umas destas subestruturas, e (LIU; YANG, 2014), que descarta a conformação parcial construída sempre que ocorre uma colisão, substituindo ela por uma cópia parcial da conformação da melhor formiga da população. Os métodos foram chamados respectivamente de *Unfold Half*, *Flexible Retrival* e *Partial Copy*, e são apresentados com mais detalhes abaixo. Cada procedimento de construção define uma versão diferente do algoritmo.

4.1.1.1 *Unfold Half*

Neste procedimento, a conformação é construída a partir de uma aresta aleatória, escolhida de forma independente para cada formiga. As extremidades da conformação são

probabilisticamente estendidas a cada iteração do algoritmo, até que os aminoácidos de índice 0 e $N-1$ sejam posicionados na malha. Mais especificamente, em uma dada iteração, uma extremidade é estendida com probabilidade igual ao número de aminoácidos ainda não posicionados naquela extremidade dividido pela soma do número de aminoácidos ainda não posicionados em ambas as extremidades.

Um aminoácido interno nunca é posicionado em uma posição da malha em que todas as posições vizinhas já estão ocupadas. Se, durante o processo de construção, um aminoácido interno não puder ser posicionado, dado que todas as posições candidatas são proibidas, o procedimento desfaz metade da distância já enovelada, recomeçando o processo de construção a partir daí. A fim de evitar que a conformação seja estendida da mesma forma, levando exatamente ao mesmo estado de colisão, a aresta a partir da qual o processo será reiniciado é proibida de adotar a direção que adotou antes.

4.1.1.2 *Flexible Retrival*

Neste procedimento, a conformação é construída a partir do meio da sequência. A conformação é dividida em duas subconformações. A estrutura que vai do aminoácido $\lfloor \frac{N}{2} \rfloor$ até o aminácido 0 define a subconformação da esquerda, enquanto que a estrutura que vai no aminoácido $\lfloor \frac{N}{2} + 1 \rfloor$ ao aminoácido $N-1$ define a conformação da direita. A conformação é estendida em ambas as direções de maneira intercalada.

Quando a conformação atinge um estado inviável, o procedimento desfaz parte de uma das duas subconformação. A subconformação na qual será aplicado o mecanismo é escolhida com base em alguns critérios:

- o mecanismo não pode ser aplicado duas vezes seguidas sobre uma mesma subconformação;
- uma subconformação só pode ser escolhida caso tenha pelo menos dois aminoácidos de comprimento;
- caso o mecanismo esteja sendo aplicado pela primeira vez, a subconformação escolhida é aquela onde ocorreu a estagnação.

O número de aminoácidos a serem removidos é escolhido aleatoriamente, respeitando o limite da subconformação, isto é, a remoção dos aminoácidos de uma subconformação nunca alcança a porção da conformação referente a outra subconformação. Além disso, cada conformação tem de ter pelo menos um aminoácido de comprimento, e por isso, os dois aminoácidos centrais da conformação nunca são removidos.

4.1.1.3 *Partial Copy*

Neste procedimento, a conformação é estendida a partir da primeira aresta da sequência. Quando um estado de colisão é atingido, a conformação parcialmente construida é descartada e substituída por uma cópia parcial da melhor conformação da população. Esta cópia tem comprimento equivalente ao comprimento da conformação descartada, de forma que a construção da aresta onde parou.

4.1.2 Fase de Busca Local

Nesta fase, cada formiga ou partícula aplica à sua conformação o método de busca por *Pull Moves*. *Pull move* (LESH; MITZENMACHER; WHITESIDES, 2003) é um conjunto de movimentos que pode reduzir a energia de uma conformação reorganizando alguns de seus aminoácidos. A seguir, define-se a implementação de um *pull move*: considere o i-ésimo aminoácido de uma proteína, localizado na posição (x_i, y_i) da malha. Seja L uma posição livre adjacente à posição (x_{i+1}, y_{i+1}), e diagonalmente adjacente à posição (x_i, y_i). Os aminoácidos das posições (x_i, y_i), (x_{i+1}, y_{i+1}) e a posição livre L formam os 3 vértices de um quadrado. Seja C o quarto vértice deste quadrado. Um *pull move* pode ser realizado se C é uma posição livre da malha ou se C é igual a (x_{i-1}, y_{i-1}). Quando C é igual a (x_{i-1}, y_{i-1}), a aplicação do *pull move* consiste em somente mover o aminoácido i para a posiçao L. Quando C é uma posição livre, o aminoácido i é movido para a posição L e o aminoácido $i-1$ é movido para posição C.

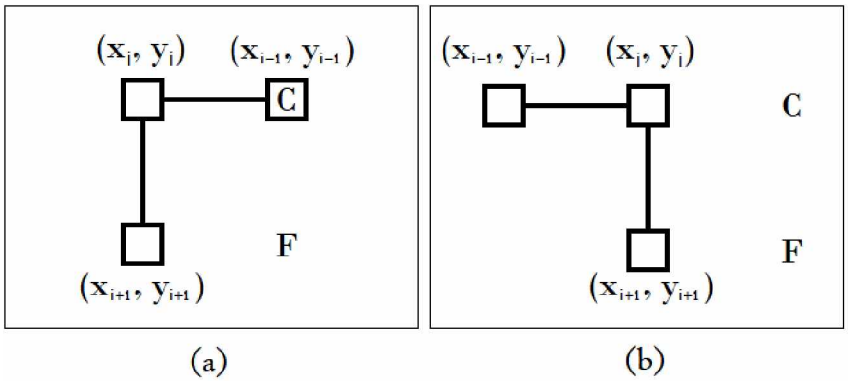

Figura 7 – Exemplo de posicionamentos adequados dos aminoácidos para aplicação do *pull move*.

Deste modo, até que uma conformação válida seja atingida, o seguinte procedimento é realizado:

Algoritmo 3: Pull Move - Correção de Conformação

1: **para** j = i - 2 até j = 0 **faça**
2: $(x_j, y_j) = (x_{j+2}, y_{j+2})$
3: **fim para**

Uma mesma conformação pode conter várias possibilidades de *pull move*, e por isso o método de busca lista todos os possíveis *pull moves* e seleciona aquele que mais decrementa a energia da conformação. Após a aplicação do *pull move*, uma nova conformação é gerada e o método de busca é repetido sobre a nova conformação. Este laço se mantém enquanto houver redução na energia.

4.1.3 Fase de Atualização do Feromônio

Nesta fase, uma parcela das formigas é selecionada para atualizar os valores de feromônio. Esta seleção é feita com base na qualidade das soluções encontradas por cada formiga, onde as formigas com as soluções de melhor qualidade são selecionadas para compôr a parcela da população responsável por atualizar o feromônio. Esta atualização é chamada de atualização elitista e é regulada por um parâmetro que indica a porcentagem de formigas que devem compôr a elite. Se este parâmetro for definido como 100%, todas as formigas participam da atualização do feromônio, e o procedimento deixa de ser elitista.

A estrutura usada para representar o feromônio é uma matriz que armazena, para cada aresta da proteína, o nível de feromônio (representado por um número real positivo) para cada um dos 3 possíveis valores que a aresta pode assumir: $\{L, R, S\}$. Tem-se uma matriz τ_{ij} de ponto flutuante, de dimensão $(n \breve{} 1)_3$, onde n é o número de aminoácidos da proteína. Ao fim da etapa de construção e da etapa de aplicação da busca por *pull moves*, cada formiga atualiza a matriz de feromônio, seguindo as Equações 4.1 e 4.2

$$\tau_{ij} = \rho\tau_{ij} + \Delta \tag{4.1}$$

$$\Delta = \frac{E(c)}{E(best)^3} \tag{4.2}$$

onde, $E(c)$ é a energia da conformação encontrada pela formiga e $E(best)$ é a energia da melhor solução encontrada pelo algoritmo até então.

O Algoritmo 4 apresenta o procedimento computacional referente ao ACO 2D HP.

4.2 Algoritmo PSO para o PSP no Modelo HP

A velocidade de uma partícula em uma dimensão i é um conjunto de tuplas do tipo $(d, p(d))$, onde $d \in \{S, L, R, F, B\}$, e $p(d)$ é a probabilidade associada ao elemento d nesta dimensão. A posição de uma partícula é um conjunto de tuplas do tipo (i, d), que associa uma direção d a uma aresta (dimensão) i, definindo uma conformação. A construção das conformações é feita de maneira análoga ao ACO. A escolha sobre qual direção uma aresta irá adotar é feita com base em uma regra probabilítica que considera

Algoritmo 4: ACO 2D HP

```
1: inicialize as trilhas de feromônio
2: para cada iteração faça
3:    para cada formiga faça
4:       construa conformação
5:    fim para
6:    para cada formiga faça
7:       aplique busca por pull moves
8:       atualize as trilhas de feromônio
9:    fim para
10: fim para
```

o número de novos contatos hidrofóbicos que a dobra irá gerar, bem como a probabilidade associada a cada direção pela velocidade da partícula.

Assim como o ACO, o fluxo principal do algoritmo pode ser definido com base em três laços aninhados. O laço mais interno itera sobre as arestas da proteína. O laço intermediário itera sobre o enxame. O laço mais externo executa até que um número máximo de iterações seja atingido.

Em seguida são redefinidos os operadores necessários para o cálculo da velocidade e posição.

- Coeficiente (c) $\times$ Velocidade (V):

$$c \times V = \{(d, p'(d)) | d \in \{S, L, R, F, B\}\} \tag{4.3}$$

$$onde, p'(d) = \begin{cases} 1, & \text{se } c \times p(d) > 1 \\ c \times p(d), & \text{caso contrário} \end{cases} \tag{4.4}$$

- Velocidade (V_1) + Velocidade (V_2):

$$V_1 + V_2 = \{(d, p'(d)) | d \in \{S, L, R, F, B\}\} \tag{4.5}$$

$$onde, p'(d) = max(p_1(d), p_2(e)) \tag{4.6}$$

- Posição (A) $-$ Posição (B):

$$A - B = \{d | d \in A \text{ e } d \notin B\} \tag{4.7}$$

- Coeficiente (c) $\times$ Posição (P)

$$c \times P = \{(d, p'(d)) | d \in \{S, L, R, F, B\}\} \tag{4.8}$$

$$onde, p'(d) = \begin{cases} 1, & \text{se } d \in P \ \& \ c > 1 \\ c, & \text{se } d \in P \ \& \ 0 \leq c \leq 1 \\ 0, & \text{se } d \notin P \end{cases} \tag{4.9}$$

A fase de construção é incorporada ao procedimento de atualização da posição e funciona da mesma maneira que a fase de construção do ACO 2D HP, exceto pela regra probabilística. A probabilidade de uma partícula escolher determinada direção d para uma aresta i é definida pela Equação 4.10:

$$p_{id} = \frac{p(d)(\eta_{id})^{\beta}}{\sum_{d \in \{S,L,R\}} p(d)(\eta_{id})^{\beta}} \tag{4.10}$$

onde $p(d)$ é a probabilidade associada à direção $d \in \{S, L, R\}$ na dimensão i e β é um valor real positivo que regula a influência da informação heurística.

Assim como no ACO, foram usados 3 diferentes procedimentos de construção, *Unfold Half*, *Flexible Retrival* e *Partial Copy*, sendo que cada um define uma versão diferente do algoritmo.

A fase de busca por *pull moves* é iniciada após todas partículas construírem suas conformações.

A seguir, o procedimento computacional referente ao PSO 2D HP:

Algoritmo 5: PSO 2D HP

```
1: inicialize a posição das partículas aleatoriamente
2: para cada iteração faça
3:    para cada partícula faça
4:       atualize a velocidade
5:       atualize a posição
6:    fim para
7:    para cada partícula faça
8:       aplique busca por pull moves
9:    fim para
10: fim para
```

5 Resultados

Na primeira seção deste capítulo são comparadas as 3 diferentes versões implementadas para o ACO. As versões são representadas por ACO_{UH} (incorpora o procedimento *Unfold Half*), ACO_{FR} (incorpora o procedimento *Flexible Retrieval*) e ACO_{PC} (incorpora o procedimento *Partial Copy*). Esta análise é feita considerando a energia das conformações geradas e o tempo médio de execução de cada versão do algoritmo. Na segunda seção, a mesma avaliação é realizada considerando as 3 diferentes versões do PSO, representadas de maneira análoga às versões do ACO, como PSO_{UH}, PSO_{FR} e PSO_{PC}. Ao final do capítulo, a melhor versão do ACO e do PSO são comparadas entre si e com resultados da literatura.

Na Tabela 1 são listadas as sequências das proteínas usadas no experimento, onde E_{best} é a menor energia conhecida para a proteína, e T é o tamanho da proteína, dado pelo número de aminoácidos.

Todos os algoritmos foram implementados na linguagem C e os experimentos foram executados em um Notebook com processador Intel® Core™ i5-7200U CPU @ 2.50GHz × 4, com 8 GB de RAM, em um ambiente Linux Ubuntu 16.04 LTS 64 bit. Os códigos-fonte implementados para o ACO e o PSO estão disponível respectivamente em https://github.com/douglas444/aco-2dhp e https://github.com/douglas444/pso-2dhp.

Tabela 1 – Tabela de *Benchmark* (SHMYGELSKA; AGUIRRE-HERNANDEZ; HOOS, 2002).

id	*Proteína*	E_{best}	T
1	$HPHP_2H_2PHP_2HPH_2P_2HPH$	-9	20
2	$H_2P_2(HP_2)_6H_2$	-9	24
3	$P_2HP_2(H_2P_4)_3H_2$	-8	25
4	$P_3H_2P_2H_2P_5H_7P_2H_2P_4H_2P_2HP_2$	-14	36
5	$P_2H(P_2H_2)_2P_5H_{10}P_6(H_2P_2)_2HP_2H_5$	-23	48
6	$H_2(PH)_3PH_4P(HP_3)_2HP_4(HP_3)_2HPH_4(PH)_3PH_2$	-21	50
7	$P_2H_3PH_8P_3H_{10}PHP_3H_{12}P_4H_6PH_2PHP$	-36	60
8	$H_{12}(PH)_2((P_2H_2)_2P_2H)_3(PH)_2H_{11}$	-42	64

5.1 Resultados para o ACO

O experimento consistiu em aplicar cada uma das três diferentes versões do ACO às proteínas do conjunto de testes. Cada versão foi executada 20 vezes sobre uma mesma proteína, sendo recuperada a energia mínima alcançada pelo algoritmo, a porcentagem de

vezes em que esta mesma energia foi alcançada dentre as 20 execuções e o tempo médio gasto na execução. Os parâmetros usados em cada versão do algoritmo são informados na Tabela 2, conforme explicados nas seções Seções 3.1 e 4.1.

Tabela 2 – Parâmetros ACO.

Parâmetro	ACO_{UH}	ACO_{FR}	ACO_{PC}
α	1	1	1
β	2	2	2
τ_0	1/3	1/3	1/3
ρ	0,2	0,2	0,2
$prob._{min}$	0,01	0,01	0
$elite(\%)$	0,1%	1%	100%
$população_{T<25}$	200	200	200
$população_{T\geq25}$	1500	1500	1500
$iterações$	500	500	500

Os resultados alcançados são informados na Tabela 3, onde E_{min} é a menor energia encontrada pelo algoritmo para a proteína, *ocorr.* é a porcentagem de ocorrência de E_{min} dentre as 20 execuções e $t_m(s)$ é a média de tempo de execução.

Tabela 3 – Resultados com ACO.

Proteína		ACO_{UH}			ACO_{FR}			ACO_{PC}		
id	E_{best}	E_{min}	*ocorr.*	$t_m(s)$	E_{min}	*ocorr.*	$t_m(s)$	E_{min}	*ocorr.*	$t_m(s)$
1	**-9**	**-9**	85%	0.90	**-9**	80%	0.60	**-9**	100%	0.95
2	**-9**	**-9**	100%	1.30	**-9**	100%	0.71	**-9**	35%	0.61
3	**-8**	**-8**	50%	1.31	**-8**	50%	0.99	**-8**	15%	1.69
4	**-14**	**-14**	80%	19.98	**-14**	35%	12.98	**-14**	55%	12.31
5	**-23**	**-23**	95%	35.51	**-23**	40%	23.04	**-23**	5%	24.09
6	**-21**	**-21**	100%	37.11	**-21**	100%	22.24	**-21**	100%	21.93
7	**-36**	**-36**	5%	58.77	-35	100%	38.45	**-36**	70%	30.02
8	**-42**	**-42**	5%	47.06	-41	25%	34.36	-39	25%	16.79

Como indicado, o ACO_{UH} foi capaz de encontrar a conformação ótima para todas as proteínas do conjunto de testes. As outras duas abordagens, entretanto, falharam em encontrar a solução ótima para as proteínas de maior comprimento. O ACO_{FR} falhou para as duas últimas proteínas, enquanto o ACO_{PC}, falhou apenas para a última proteína. Apesar do ACO_{PC} ter alcançado a solução ótima para um número maior de proteínas, para a proteína de maior comprimento, a solução subótima gerada se afastou da solução ótima em 3 unidades de energia, enquanto que o ACO_{FR} se afastou da solução ótima para última proteína por somente 1 unidade.

Quanto ao tempo médio de execução, o ACO_{PC} obteve os melhores resultados. O ACO_{FR} ocupou a segunda posição, enquanto que o ACO_{UH} foi a versão mais lenta do

algoritmo. Uma possível explicação para o desempenho superior do ACO_{PC} quanto ao tempo médio de execução está no fato de que o ACO_{FR} e o ACO_{UH} reconstroem parte da conformação sempre que ocorre uma colisão, enquanto que o ACO_{PC} apenas substitui a subconformação inválida por uma subconformação válida já construída.

A versão ACO_{UH} foi selecionada como a versão com melhor desempenho, quando considerada a capacidade de encontrar a solução ótima para todas as proteínas do conjunto de testes.

5.2 Resultados para o PSO

O experimento para o PSO foi realizado de maneira análoga ao ACO, com as 3 versões do algoritmo sendo executadas 20 vezes de maneira independente para cada uma das instâncias da base testes. Os parâmetros usados em cada versão do PSO são informados na Tabela 4, descritos nas seções 3.2 e 4.1. Os resultados alcançados são informados na Tabela 5.

Tabela 4 – Parâmetros PSO.

Parâmetro	PSO_{UH}	PSO_{FR}	PSO_{PC}
c_1	2,1	2,1	2,1
c_2	2,1	2,1	2,1
w	0,5	0,5	0,5
β	2	2	2
$prob._{min}$	0,01	0,01	0
$população_{T<25}$	200	200	200
$população_{T\geq25}$	1500	1500	1500
iterações	500	500	500

Tabela 5 – Resultados com PSO.

Proteína		PSO_{UH}			PSO_{FR}			PSO_{PC}		
id	E_{best}	E_{min}	*ocorr.*	$t_m(s)$	E_{min}	*ocorr.*	$t_m(s)$	E_{min}	*ocorr.*	$t_m(s)$
1	**-9**	**-9**	100%	1.10	**-9**	100%	1.11	**-9**	100%	0.95
2	**-9**	**-9**	100%	1.64	**-9**	100%	1.62	**-9**	95%	0.61
3	**-8**	**-8**	75%	1.83	**-8**	100%	1.99	**-8**	30%	1.69
4	**-14**	**-14**	100%	21.23	**-14**	95%	21.39	**-14**	85%	12.31
5	**-23**	**-23**	50%	42.51	-22	30%	33.14	**-23**	65%	24.09
6	**-21**	**-21**	95%	39.82	**-21**	15%	44.21	**-21**	90%	21.93
7	**-36**	**-36**	20%	66.26	-35	90%	44.87	**-36**	25%	30.02
8	**-42**	**-42**	40%	67.41	-37	5%	52.92	**-42**	25%	16.79

Como indicado, o PSO_{UH} e o PSO_{PC}, foram ambos capazes de encontrar a conformação ótima para todas as proteínas do conjunto de testes. Apesar disso, o PSO_{UH}

conseguiu um número maior de ocorrências das soluções ótimas em suas execuções, perdendo para o ACO_{PC} apenas nas instâncias 5 e 7. O PSO_{FR} perdeu para a quinta, sétima e oitava proteína.

Quanto ao tempo médio de execução, o PSO_{PC} alcançou os melhores resultados, conseguindo o menor tempo médio de execução para todas as instâncias, sendo que para a instância 8, o algoritmo conseguiu um tempo médio 4 vezes menor que o PSO_{FR} e 3 vezes menor que o PSO_{UH}, repetindo um resultado de tempo semelhante ao ACO.

A versão PSO_{UH} foi selecionada como a versão com melhor desempenho, pois além de encontrar a solução ótima para todas as instâncias, alcançou uma porcentagem maior de ocorrências das soluções ótimas.

5.3 Comparação do ACO com o PSO

A Tabela 6 compara o ACO_{UH} e o PSO_{UH}.

Tabela 6 – Comparação do ACO_{UH} e PSO_{UH}.

Proteína		ACO_{UH}			PSO_{UH}		
id	E_{best}	E_{min}	*ocorr.*	$t_m(s)$	E_{min}	*ocorr.*	$t_m(s)$
1	**-9**	**-9**	80%	0.90	**-9**	100%	1.10
2	**-9**	**-9**	100%	1.30	**-9**	100%	1.64
3	**-8**	**-8**	50%	1.31	**-8**	75%	1.83
4	**-14**	**-14**	80%	19.98	**-14**	100%	21.23
5	**-23**	**-23**	95%	35.51	**-23**	50%	42.51
6	**-21**	**-21**	100%	37.11	**-21**	95%	39.82
7	**-36**	**-36**	5%	58.77	**-39**	20%	66.26
8	**-42**	**-42**	5%	47.06	**-42**	40%	67.41

Como indicado na Tabela 6, ambos os algoritmos alcançaram a solução ótima para todas as instâncias do conjunto. A diferença está evidenciada na porcentagem de ocorrências das soluções. Com exceção das instâncias 5 e 6, o PSO_{UH} conseguiu uma taxa maior de ocorrências. Quanto ao tempo de execução, entretanto, o ACO_{UH} mostrou-se melhor, alcançando o menor tempo médio de execução para todas as instâncias do conjunto de testes.

Devido ao número maior de ocorrências das soluções ótimas, o algoritmo PSO_{UH} foi selecionado como o algoritmo de melhor desempenho.

5.4 Comparação do ACO e do PSO com resultados da literatura

A fim de validar o ACO_{UH} e o PSO_{UH} como propostas viáveis para o tratamento do PSP no modelo 2D HP, foi realizada uma comparação dos resultados alcançados por

ambas as abordagens com algoritmos da literatura. A Tabela 7 compara os valores de energia obtidos pelos algoritmos ACO_{UH} e PSO_{UH} e os algoritmos da literatura Otimização por Colônia de Formiga Heurístico (HACO, do inglês *Heuristic Ant Colony Optimization*) (LIU; YANG, 2014), Sistema Imunológico Artificial (IA, do inglês *Immune Algorithm*) (CUTELLO et al., 2007), Monte Carlo Evolucionário (EMC, do inglês *Evolutionary Monte Carlo*) (LIANG; WONG, 2001) e Algoritmo Genético (GA, do inglês *Genetic Algorithm*) (UNGER; MOULT, 1993). O HACO é um algoritmo baseado na metaheurística ACO que incorpora o método de busca por *pull move* e que utiliza o procedimento de construção *Partial Copy*. Apesar de se assemelhar com a abordagem ACO_{PC} implementada neste trabalho, o HACO utiliza informação prévia da energia ótima de uma proteína, isto é, a energia ótima conhecida para a proteína é fornecida ao algoritmo de modo a auxiliar no processo de busca.

De acordo com seus respectivos trabalhos, os algoritmos IA, EMC e GA foram executados 30 vezes sobre cada instâncias, sendo recuperada a energia mínima alcançada para cada uma. O HACO, por sua vez, foi executado 20 vezes para cada instâncias.

Tabela 7 – Comparação do ACO_{UH} e PSO_{UH} com resultados da literatura.

Proteína	E_{best}	ACO_{UH}	PSO_{UH}	$HACO$	IA	EMC	GA
1	**-9**	**-9**	**-9**	**-9**	**-9**	**-9**	**-9**
2	**-9**	**-9**	**-9**	**-9**	**-9**	**-9**	**-9**
3	**-8**	**-8**	**-8**	**-8**	**-8**	**-8**	**-8**
4	**-14**	**-14**	**-14**	**-14**	**-14**	**-14**	**-14**
5	**-23**	**-23**	**-23**	**-14**	**-23**	**-23**	**-23**
6	**-21**	**-21**	**-21**	**-21**	**-21**	**-21**	**-21**
7	**-36**	**-36**	**-36**	**-36**	-35	-34	-35
8	**-42**	**-42**	**-42**	-41	-39	-37	-39

Como indicado, os algoritmos ACO_{UH} e ACO_{UH} foram os únicos a encontrar a solução ótima para todas as proteínas do conjunto de testes.

6 Conclusão

Neste trabalho foi realizada uma comparação de desempenho, tanto do ponto de vista computacional quanto da qualidade das soluções, entre os algoritmos ACO e PSO com o método de busca por *pull move* na predição de estrutura de proteínas no Modelo 2D HP. Ambos os algoritmos foram analisados a partir de 3 diferentes abordagens, onde cada uma adotou um dos seguintes métodos de construção de conformação de proteínas: *Unfold Half* (SHMYGELSKA; AGUIRRE-HERNANDEZ; HOOS, 2002) (Seção 4.1.1.3), *Flexible Retrival* (HU; ZHANG; LI, 2008) (Seção 4.1.1.2) e *Partial Copy* (LIU; YANG, 2014) (Seção 4.1.1.3).

Na comparação entre as diferentes abordagens, os resultados evidenciaram que os algoritmos do ACO e do PSO baseados no procedimento de construção *Unfold Half*, chamados de ACO_{UH} e PSO_{UH}, são as melhores abordagens para ambos os algoritmos, quando considerada a qualidade das soluções. Vale lembrar o funcionamento do procedimento *Unfold Half*: as extremidades da conformação são probabilisticamente estendidas a cada iteração do algoritmo, e sempre que um estado de colisão é alcançado, o procedimento desfaz metade da distância já enovelada, recomeçando o processo de construção a partir deste ponto. Para o ACO, a abordagem ACO_{UH} foi a única a alcançar a solução ótima para todas as instâncias da base. Para o PSO, tanto a abordagem PSO_{UH} quanto a abordagem PSO_{PC}, baseada no procedimento de construção *Partial Copy*, foram capazes de encontrar a solução ótima para todas as instâncias da base, mas a abordagem PSO_{UH} conseguiu alcançar uma taxa maior de ocorrências da solução ótima dentro das 20 execuções realizadas sobre cada proteína.

Na comparação entre o ACO e o PSO, neste trabalho os resultados evidenciaram que o PSO é o melhor algoritmo para tratamento do problema em questão. Apesar de ambos os algoritmos alcançarem a solução ótima para todas as instâncias do conjunto de testes, o PSO alcançou um taxa maior de ocorrências das soluções ótimas.

Na comparação entre o ACO e PSO com algoritmos da literatura, concluiu-se que ambos os algoritmos, ACO e PSO, são métodos promissores para tratamento do problema, uma vez que os algoritmos foram os únicos a alcançar a solução ótima para todas as instâncias do conjunto de teste. Possíveis trabalhos futuros incluem:

- uma análise sobre a influência dos parâmetros de cada algoritmo sobre o tempo de execução e a qualidade das soluções considerando as diferentes abordagens para a construção das conformações;
- a adaptação dos algoritmos ACO e PSO com busca por *pull move* para o modelo

3D HP;

- a análise da convergência da população de partícula, para o PSO, e de formigas, para ACO, por meio de métodos para cálculo da distância entre conformações;
- o desenvolvimento de critérios de paradas mais eficientes;
- a comparação do método de busca por *pull move* com outros métodos de busca local.

Referências

BĂUTU, A.; LUCHIAN, H. Protein structure prediction in lattice models with particle swarm optimization. In: SPRINGER. *International Conference on Swarm Intelligence*. [S.l.], 2010. p. 512–519. Citado 3 vezes nas páginas 11, 12 e 25.

BRASIL, C. R. S. *Algoritmo evolutivo de muitos objetivos para predição ab initio de estrutura de proteínas*. Tese (Doutorado) — Universidade de São Paulo, 2012. Citado 4 vezes nas páginas 11, 12, 19 e 20.

CHEN, W.-N. et al. A novel set-based particle swarm optimization method for discrete optimization problems. *IEEE Transactions on evolutionary computation*, Citeseer, v. 14, n. 2, p. 278–300, 2010. Citado na página 26.

CITROLO, A. G.; MAURI, G. A hybrid monte carlo ant colony optimization approach for protein structure prediction in the hp model. *arXiv preprint arXiv:1309.7690*, 2013. Citado na página 12.

CRESCENZI, P. et al. On the complexity of protein folding. *Journal of computational biology*, v. 5, n. 3, p. 423–465, 1998. Citado na página 11.

CUTELLO, V. et al. An immune algorithm for protein structure prediction on lattice models. *IEEE transactions on evolutionary computation*, IEEE, v. 11, n. 1, p. 101–117, 2007. Citado na página 38.

DILL, K. A. et al. Principles of protein folding—a perspective from simple exact models. *Protein science*, Wiley Online Library, v. 4, n. 4, p. 561–602, 1995. Citado na página 20.

DORIGO, M. Optimization, learning and natural algorithms. *PhD Thesis, Politecnico di Milano*, 1992. Citado 2 vezes nas páginas 11 e 22.

DORIGO, M.; MANIEZZO, V.; COLORNI, A. Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, IEEE, v. 26, n. 1, p. 29–41, 1996. Citado 2 vezes nas páginas 22 e 23.

DOUDNA, J.; COX, M. Biologia molecular-princípios e técnicas. 2012. Citado 7 vezes nas páginas 4, 10, 11, 12, 15, 17 e 18.

EBERHART, R.; KENNEDY, J. A new optimizer using particle swarm theory. In: IEEE. *Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on*. [S.l.], 1995. p. 39–43. Citado 2 vezes nas páginas 11 e 24.

ENGELBRECHT, A. P. *Fundamentals of Computational Swarm Intelligence*. USA: John Wiley & Sons, Inc., 2006. ISBN 0470091916. Citado na página 11.

FIDANOVA, S.; LIRKOV, I. Ant colony system approach for protein folding. In: IEEE. *Computer Science and Information Technology, 2008. IMCSIT 2008. International Multiconference on*. [S.l.], 2008. p. 887–891. Citado 2 vezes nas páginas 12 e 24.

HU, X.-M.; ZHANG, J.; LI, Y. Flexible protein folding by ant colony optimization. In: *Computational Intelligence in Biomedicine and Bioinformatics*. [S.l.]: Springer, 2008. p. 317–336. Citado 4 vezes nas páginas 12, 24, 28 e 39.

JANA, N. D.; SIL, J. Particle swarm optimization with backtracking in protein structure prediction problem. In: IEEE. *Signal Processing, Communication and Computing (ICSPCC), 2012 IEEE International Conference on*. [S.l.], 2012. p. 734–738. Citado 2 vezes nas páginas 12 e 26.

KONAR, A. *Computational Intelligence: Principles, Techniques and Applications*. Berlin, Heidelberg: Springer-Verlag, 2005. ISBN 3540208984. Citado na página 11.

LAU, K. F.; DILL, K. A. A lattice statistical mechanics model of the conformational and sequence spaces of proteins. *Macromolecules*, ACS Publications, v. 22, n. 10, p. 3986–3997, 1989. Citado 3 vezes nas páginas 11, 12 e 20.

LESH, N.; MITZENMACHER, M.; WHITESIDES, S. A complete and effective move set for simplified protein folding. In: ACM. *Proceedings of the seventh annual international conference on Research in computational molecular biology*. [S.l.], 2003. p. 188–195. Citado 2 vezes nas páginas 27 e 30.

LIANG, F.; WONG, W. H. Evolutionary monte carlo for protein folding simulations. *The Journal of Chemical Physics*, AIP, v. 115, n. 7, p. 3374–3380, 2001. Citado na página 38.

LIN, C.-J.; SU, S.-C. Protein 3d hp model folding simulation using a hybrid of genetic algorithm and particle swarm optimization. *International Journal of Fuzzy Systems*, v. 13, n. 2, 2011. Citado 2 vezes nas páginas 12 e 25.

LIU, Z.; YANG, Z. Heuristic ant colony optimization algorithm for predicting the structures of 2d hp model proteins. In: IEEE. *Biomedical Engineering and Informatics (BMEI), 2014 7th International Conference on*. [S.l.], 2014. p. 719–723. Citado 5 vezes nas páginas 12, 24, 28, 38 e 39.

LLANES, A. et al. Parallel ant colony optimization for the hp protein folding problem. In: SPRINGER. *International Conference on Bioinformatics and Biomedical Engineering*. [S.l.], 2016. p. 615–626. Citado 3 vezes nas páginas 12, 22 e 24.

MANSOUR, N.; KANJ, F.; KHACHFE, H. Particle swarm optimization approach for protein structure prediction in the 3d hp model. *Interdisciplinary Sciences: Computational Life Sciences*, Springer, v. 4, n. 3, p. 190–200, 2012. Citado 2 vezes nas páginas 12 e 26.

NELSON, D. L.; COX, M. M. *Lehninger: principios de bioquímica*. [S.l.: s.n.], 2015. Citado 9 vezes nas páginas 4, 10, 14, 15, 16, 17, 18, 19 e 20.

NESHAT, M. et al. Artificial fish swarm algorithm: a survey of the state-of-the-art, hybridization, combinatorial and indicative applications. *Artificial intelligence review*, Springer, v. 42, n. 4, p. 965–997, 2014. Citado na página 24.

RAN, W.; LIU, L.; YANG, G. A hybrid ant colony algorithm for vehicle routing problem with time windows. *Information Technology Journal*, Asian Network for Scientific Information (ANSINET), v. 12, n. 20, p. 5701–5706, 2013. Citado 2 vezes nas páginas 22 e 24.

SHI, Y.; EBERHART, R. A modified particle swarm optimizer. In: IEEE. *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*. [S.l.], 1998. p. 69–73. Citado na página 25.

SHMYGELSKA, A.; AGUIRRE-HERNANDEZ, R.; HOOS, H. H. An ant colony optimization algorithm for the 2d hp protein folding problem. In: SPRINGER. *International Workshop on Ant Algorithms*. [S.l.], 2002. p. 40–52. Citado 7 vezes nas páginas 5, 11, 12, 24, 28, 34 e 39.

THILAGAVATHI, N.; AMUDHA, T. Rank based ant algorithm for 2d-hp protein folding. In: *Computational Intelligence in Data Mining-Volume 3*. [S.l.]: Springer, 2015. p. 441–451. Citado 2 vezes nas páginas 12 e 24.

UNGER, R.; MOULT, J. Genetic algorithms for protein folding simulations. *Journal of molecular biology*, Elsevier, v. 231, n. 1, p. 75–81, 1993. Citado na página 38.

WANG, S. Improved swarm intelligence algorithm for protein folding prediction. *Cluster Computing*, Springer, p. 1–10, 2018. Citado na página 24.

XIAO, J.; LI, L.-P.; HU, X.-M. Solving lattice protein folding problems by discrete particle swarm optimization. *Journal of Computers*, v. 9, n. 8, 2014. Citado 2 vezes nas páginas 12 e 26.

YAN, G.; SHI, B. The application of ant colony optimization algorithm in dna encoding. *Journal of Computational Information Systems*, v. 7, p. 3591–3598, 2011. Citado na página 22.

YANG, C.-H. et al. Hybrid high exploration particle swarm optimization algorithm improves the prediction of the 2-dimensional hydrophobic-polar model for protein folding. *Current Bioinformatics*, Bentham Science Publishers, v. 13, n. 2, p. 182–192, 2018. Citado 2 vezes nas páginas 12 e 26.