

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Júlia Manfrin Dias

**Comparação do Algoritmo de Colônia de
Formiga e Algoritmo Evolutivo Aplicado ao
problema de predição de estruturas de proteínas**

Uberlândia, Brasil

2016

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Júlia Manfrin Dias

**Comparação do Algoritmo de Colônia de Formiga e
Algoritmo Evolutivo Aplicado ao problema de predição de
estruturas de proteínas**

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: Professora *Dr^a*. Christiane Regina Soares Brasil

Universidade Federal de Uberlândia – UFU

Faculdade de Ciência da Computação

Bacharelado em Ciência da Computação

Uberlândia, Brasil

2016

Júlia Manfrin Dias

Comparação do Algoritmo de Colônia de Formiga e Algoritmo Evolutivo Aplicado ao problema de predição de estruturas de proteínas

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, 16 de dezembro de 2016:

**Professora *Dr^a*. Christiane Regina
Soares Brasil**
Orientadora

**Professora *Dr^a*. Maria Adriana Vidigal
de Lima**

**Professor Dr. Paulo Henrique Ribeiro
Gabriel**

Uberlândia, Brasil
2016

Dedico aos meus pais, Darlene e Rafael.

Agradecimentos

Agradeço aos meus pais Darlene e Rafael, pelo incentivo que sempre me deram, ao apoio nas situações mais difíceis e por toda compreensão.

Agradeço à minha família, tias e primos, pelo apoio que me deram.

Agradeço aqueles colegas que me ajudaram, de alguma forma, ao longo do curso.

Agradeço à Professora Christiane por toda paciência, sugestões, ajudas e críticas.

“Cair sete vezes, levantar oito.”

Provérbio Japonês

Resumo

Métodos de otimização são amplamente usados em diversas áreas científicas para encontrar soluções de problemas complexos, classificados como problemas não polinomiais (NP). Os métodos de otimização estudados neste trabalho são o Algoritmo Evolutivo (AE) e a Otimização por Colônia de Formiga (ACO – *Ant Colony Optimization*). Esses algoritmos são bioinspirados, isto é, se baseiam em processos que ocorrem na natureza. Ambos os métodos, AE e o ACO, são aplicados para encontrar um conjunto de soluções para o problema de predição de proteínas (PSP – *Protein Structure Problem*), caracterizado como um problema NP por sua alta complexidade. O problema PSP trata-se da busca de estruturas tridimensionais de proteínas, cujas estruturas estão diretamente relacionadas à funcionalidade das mesmas, justificando a grande importância dessa área de pesquisa. Conhecendo-se as funções de uma proteína, pode-se aplicá-las na obtenção e no estudo de novas soluções para doenças graves. Portanto, este trabalho tem como objetivo uma comparação entre esses dois métodos aplicados ao PSP. Os resultados foram comparados entre si, tanto do ponto de vista computacional quanto bioquímico. O ACO obteve um resultado melhor que o AE para algumas proteínas testadas.

Palavras-chave: Otimização, Algoritmo Evolutivo, Otimização por Colônia de Formigas, Predição de Estrutura de Proteína .

Lista de ilustrações

Figura 1 – Cruzamento entre dois pais.	13
Figura 2 – Mutação em um indivíduo.	13
Figura 3 – Estrutura de um aminoácido.	18
Figura 4 – Representação de estruturas das proteínas	20
Figura 5 – Representação de estrutura de uma proteína no modelo HP.	22
Figura 6 – Deslocamento da formiga.	28
Figura 7 – Sentidos de movimentos da formiga.	29
Figura 8 – Formiga com orientação na vertical.	29
Figura 9 – Formiga com orientação na horizontal.	29
Figura 10 – Cálculo do valor heurístico.	31
Figura 11 – Representação dos genes.	33
Figura 12 – Alocação dos aminoácidos pelo AE.	34
Figura 13 – Dobramento da proteína feita pelo AE.	34
Figura 14 – <i>Crossover</i> 1-ponto.	35
Figura 15 – <i>Crossover</i> 2-pontos.	36
Figura 16 – Implementação da mutação.	36
Figura 17 – Proteínas utilizadas para obter os resultados.	39
Figura 18 – Estrutura ótima da proteína de 20 aminoácidos.	41
Figura 19 – Estruturas ótimas encontradas pelo ACO.	41
Figura 20 – Estruturas ótimas encontradas pelo AE.	42
Figura 21 – Estrutura ótima da proteína de 24 aminoácidos.	43
Figura 22 – Estrutura ótima obtida na execução do ACO.	44
Figura 23 – Estrutura ótima obtida na execução do AE.	44

Lista de abreviaturas e siglas

ACO	<i>Ant Colony Optimization</i>
ACS	<i>Ant Colony system</i>
AE	Algoritmo Evolutivo
AG	Algoritmo Genético
AS	<i>Ant System</i>
MMAS	<i>Min-Max Ant System</i>
PSP	<i>Protein Structure Problem</i>
RNM	<i>Ressonância Nuclear Magnética</i>

Sumário

1	INTRODUÇÃO	10
1.1	Objetivos	11
1.2	Justificativa	11
2	ALGORITMOS DE OTIMIZAÇÃO	12
2.1	Algoritmos Evolutivos	12
2.2	<i>Ant Colony Optimization</i> - ACO	14
3	PROBLEMA DE PREDIÇÃO DE ESTRUTURAS DE PROTEÍNA	18
3.1	Estruturas das Proteínas	18
3.2	Métodos para predição de estrutura	20
3.3	Modelo HP para o problema PSP	21
4	TRABALHOS CORRELATOS	24
5	IMPLEMENTAÇÃO DO TRABALHO	27
5.1	Descrição do método ACO para o PSP	27
5.1.1	Representação computacional da formiga	27
5.1.2	Movimentação da formiga	28
5.1.3	Cálculo da probabilidade	30
5.1.4	Matriz de feromônios	31
5.1.5	Atualização do feromônio	31
5.2	Descrição do método AE para o PSP	33
5.2.1	Representação computacional do indivíduo	33
5.2.2	Formação da estrutura bidimensional da proteína no AE	33
5.2.3	Operações de <i>crossover</i> , mutação e seleção dos pais	35
6	RESULTADOS E DISCUSSÕES	38
6.1	Resultados com a proteína de 20 aminoácidos	39
6.2	Resultados com a proteína de 24 aminoácidos	42
6.3	Resultados com a proteína de 25 aminoácidos	44
6.4	Resultados com a proteína de 36 aminoácidos	45
6.5	Resultados com a proteína de 48 aminoácidos	46
7	CONCLUSÃO	49
	Referências	51

1 Introdução

Existem diversos problemas complexos no mundo real, onde não é possível ser obtida uma solução viável a partir de abordagens tradicionais. Como exemplo, podem ser citados os problemas de distribuição elétrica de uma cidade, roteamento de rede, roteirização de transporte, entre tantos outros. Percebe-se que eles se caracterizam como problemas combinatórios, ou seja, que exigem uma busca da melhor solução em um espaço muito vasto onde há todas as combinações das possíveis soluções.

Um dos problemas complexos mais desafiadores da atualidade é o problema de predição de estruturas de proteínas (do inglês, *Protein Structure Problem - PSP*), que também é um problema combinatório. Há um grande conjunto de possibilidades de conformação tridimensional de uma proteína, mas dentre estas, poucas são, de fato, conformações estáveis¹ da proteína. Vale lembrar que as proteínas são moléculas biológicas fundamentais para a vida na Terra, sendo cada vez mais necessário o entendimento das estruturas 3D estáveis dessas moléculas, pois as estruturas das mesmas estão diretamente relacionadas às suas características funcionais. Portanto, conhecendo e entendendo as funções das proteínas, podem-se gerar novas drogas ou remédios para prevenção de doenças, até hoje consideradas incuráveis.

Neste contexto, métodos computacionais de otimização estão tratando desse problema, posto que os métodos convencionais (RNM e Cristalografia) não são eficientes em termo de tempo e custo. A otimização é o processo que viabiliza o melhor rendimento para realizar alguma tarefa, seja ela computacional ou não. No contexto computacional, a otimização pode ser representada por um algoritmo que seja mais eficiente e apresente soluções ótimas ou aproximações das mesmas a um determinado problema complexo. Cabe ressaltar que muitas vezes não é obtida uma única solução, e nesses casos, pode ser difícil definir qual a melhor solução no conjunto das possíveis (HAUPT; HAUPT, 2004).

Há vários métodos de otimização para encontrar soluções de problemas NP, ou seja, não polinomiais. Os métodos estudados neste projeto foram o Algoritmo Evolutivo e a Otimização por Colônia de Formiga. Esses métodos são bioinspirados, pois se baseiam em processos que ocorrem na natureza.

Os Algoritmos Evolutivos (AEs) são uma das técnicas de otimização de problemas. São baseados em teorias evolucionistas de Darwin, como a Seleção Natural. Também é inspirado no processo de reprodução como ocorre na natureza. Foram desenvolvidos por Holland (1975) e popularizados por Goldberg (1989), entre os anos 1960 e 1970.

A Otimização por Colônia de Formigas (do inglês, *Ant Colony Optimization* –

¹ Estruturas estáveis são aquelas com energia mínima.

ACO) é um método de otimização de problemas baseado no comportamento de formigas na busca de alimentos. Essa heurística foi desenvolvida por Marco Dorigo na década de 1990 (DORIGO; GAMBARDILLA, 1997). Na natureza as formigas percorrem vários caminhos da colônia até o local onde há o alimento e vão depositando feromônio ao longo dele, formando trilhas. As demais formigas da colônia seguem as trilhas de feromônios e localizam o caminho mais curto (CARVALHO; RAMOS, 2007).

1.1 Objetivos

O objetivo principal deste trabalho foi realizar um estudo comparativo entre duas técnicas de otimização computacional para o problema de predição de estruturas de proteínas usando o modelo de representação Hidrofóbico-Polar (HP). Este modelo simplifica as estruturas das proteínas, classificando os aminoácidos em dois grupos: hidrofóbicos (H) ou polares (P). Os aminoácidos insolúveis na água ou hidrofóbicos ficam no grupo H e os polares integram o grupo P. As técnicas estudadas foram o Algoritmo Evolutivo e a Otimização por Colônia de Formigas. Os resultados foram comparados por meio de métricas de avaliação de predições, sob o ponto de vista computacional e bioquímico.

1.2 Justificativa

É importante investigar os métodos de otimização voltados para o problema de predição, a fim de analisar as possibilidades de encontrar soluções adequadas, em um tempo viável. Neste trabalho, serão comparados dois métodos, e por meio dos resultados, pode-se ter um direcionamento de qual situação (parâmetros) cada método teve melhor desempenho, considerando, desse modo, não apenas o custo computacional, mas também a qualidade das soluções.

2 Algoritmos de Otimização

Os algoritmos de otimização têm o objetivo de melhorar a busca pela solução de um problema considerado complexo, encontrando a melhor solução possível para resolvê-lo ou aproximações para mesma. São muito utilizados para encontrar possíveis respostas aos problemas NP¹. Para este trabalho foram estudados dois destes algoritmos, o Algoritmo Evolutivo e a Otimização por Colônia de Formigas, ambos inspirados em processos naturais. A seguir, serão descritos estes dois algoritmos que foram desenvolvidos neste trabalho.

2.1 Algoritmos Evolutivos

Os Algoritmos Evolutivos (AEs) são uma das técnicas de otimização de problema inspiradas em teorias evolucionistas de Darwin, como a Seleção Natural. Estes algoritmos foram inicialmente desenvolvidos por [Holland \(1975\)](#) e popularizados por [Goldberg \(1989\)](#), entre os anos 1960 e 1970.

Na biologia, um indivíduo representa um ser vivo e a população é um conjunto de indivíduos da mesma espécie que vivem no mesmo lugar. Desse modo, o processo de evolução consiste no nascimento de novos indivíduos, que possuem uma mudança nas características herdadas de seus pais, ou de gerações anteriores. Essas mudanças ocorrem por meio da reprodução entre os seres vivos e das mutações gênicas que eles podem sofrer.

A mutação é uma alteração em um ou mais genes de um cromossomo, enquanto que a reprodução ocorre por trocas de segmentos entre cromossomos. A partir de um breve conhecimento dos termos biológicos, é possível entender o funcionamento do AE. No ambiente da computação, um indivíduo será representado por um vetor, por exemplo, e uma população pode ser entendida como vários vetores ou uma matriz.

O AE denomina o indivíduo (vetor) como um cromossomo e cada posição pode ser interpretada como um gene. O indivíduo representa uma possível solução para problema, que pode ser satisfatória ou não. O algoritmo começa a execução gerando aleatoriamente uma população inicial de indivíduos, com as possíveis soluções. Esses indivíduos passam por uma avaliação de sua suposta solução e recebem um valor que indica o quão boa é essa solução para o problema. A cada geração do AE alguns indivíduos da população são escolhidos e por meio das operações de cruzamento (*crossover*) a população se renova.

¹ Problema NP significa não polinomial. Esse termo é usado para se referir a problemas cuja solução não é obtida em tempo polinomial, ou seja, as respostas para estes problemas levam muito tempo para serem computadas, muitas vezes são solucionadas em questão de anos, centenas de anos ou até milênios.

Imitando a natureza, a mutação ocorre em menores proporções na população.

Portanto, as operações de reprodução que ocorrem no AE são:

- *Crossover*: é a troca de genes entre os cromossomos dos pais. O descendente recebe parte dos genes de um pai e outra parte do outro pai, tendo assim características parecidas aos dois. No exemplo da Figura 1 a seguir, é mostrado o cruzamento de dois indivíduos, onde a primeira metade de um pai e a segunda metade do outro pai são enviados para os filhos. Desse modo, os filhos contêm informações advindas dos pais.

Cruzamento (*Crossover*)

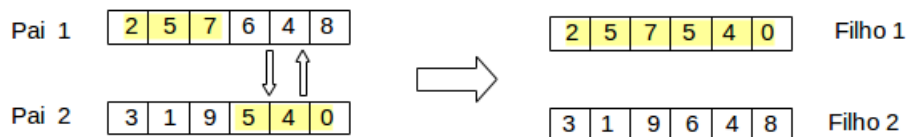


Figura 1 – Cruzamento entre dois pais.

- *Mutação*: consiste na troca de dois ou mais genes do cromossomo. De acordo com a Figura 2, um gene pode ser escolhido para sofrer a mutação ou dois genes são escolhidos e trocam seus valores. Essa mudança gera características novas na população, que podem ser boas ou não. Caso sejam consideradas boas, o novo indivíduo é mantido para as próximas gerações. Quando esta mutação não é satisfatória para a população, esta é descartada. Esse operador é ilustrado no exemplo a seguir.

Mutação



Figura 2 – Mutação em um indivíduo.

Assim que os descendentes são gerados, suas possíveis soluções são avaliadas por uma função objetivo (*fitness*) e então eles são inseridos na população. Essa função objetivo depende especificamente de cada problema a ser tratado. Os indivíduos que não apresentem uma boa avaliação são substituídos pelos filhos. Esse processo é realizado a cada geração e repetido até que um critério de parada seja atingido, que pode ser a resposta

ótima alcançada ou um dado número de gerações pré-definido. A seguir um pseudocódigo do algoritmo evolutivo (Algorithm 1).

Algorithm 1 Pseudocódigo do algoritmo evolutivo

- 1: Gerar população inicial.
 - 2: Avaliar população segundo a função objetivo.
 - 3: **enquanto** número de gerações não atingido **faça**
 - 4: Selecionar indivíduos para reprodução.
 - 5: Realizar cruzamento entre os indivíduos.
 - 6: Realizar mutação em alguns indivíduos.
 - 7: Inserir filhos na população.
 - 8: Avaliar população segundo a função objetivo.
 - 9: **fim enquanto**
-

2.2 *Ant Colony Optimization - ACO*

Desenvolvidas por [Dorigo e Gambardella \(1997\)](#), a Otimização por Colônia de Formigas (do inglês, *Ant Colony Optimization - ACO*), se baseia em como as formigas buscam seus alimentos. Na natureza, as formigas saem do formigueiro e inicialmente fazem caminhos aleatórios para encontrar alimento. A formação de um único caminho que conduz a maioria das formigas até o local do alimento é realizado devido a comunicação que elas têm entre si por meio de uma substância química. Essa substância química, chamada feromônio, é o que as formigas depositam no chão por onde passam, formando as trilhas de feromônio ([DORIGO; BLUM, 2005](#)). Uma vez que estes insetos identificam essa substância há a comunicação com o restante da colônia, e assim as demais formigas seguem o mesmo caminho. Esse caminho geralmente é o menor encontrado entre a colônia e o local do alimento.

Observando esse comportamento, surgiram os estudos sobre a possibilidade de se replicar computacionalmente, algo igual ou parecido, visto que as formigas naturalmente encontravam caminhos ótimos. Assim surgiu o ACO, na década de 1990. O algoritmo ACO faz uso de recursos que mimetizam as formigas e o feromônio. São geradas soluções probabilísticas baseadas na informação heurística. A formiga gera uma possível solução efetuando um cálculo de probabilidade, o qual determina o caminho que deve seguir. São consideradas a quantidade de feromônio existente na trilha e a distância para o cálculo de probabilidade. Depois de reconhecido o caminho, a informação do feromônio é atualizada, considerando o depósito das demais formigas que passaram pelo caminho e a evaporação desta substância.

Existem diferentes variações do ACO criado por Dorigo. As principais delas são o ACS (*Ant Colony System*), AS (*Ant System*) e MMAS (*Max-Min Ant System*). Neste trabalho foi utilizada a abordagem MMAS.

Essas variações se diferem entre si pela atualização do feromônio. O AS aborda praticamente todo o conceito do ACO, tendo seus valores de feromônios atualizados por todas as formigas da colônia. As fórmulas para o cálculo de probabilidade e da atualização do feromônio são apresentadas abaixo:

O cálculo da probabilidade de ir para o próximo local é definido a seguir:

$$P_{ij} = (\tau_{ij})^\alpha (\eta_{ij})^\beta / \sum_{e \in N} (\tau_{ie})^\alpha (\eta_{ie})^\beta \quad (2.1)$$

Onde:

- i e j significam, respectivamente o ponto em que a formiga está e o ponto que a formiga pode ir.
- τ_{ij} é o valor do feromônio no local pretendido.
- η_{ij} é o valor da informação heurística.
- As letras α e β representam o quão importante será o feromônio e a informação heurística, respectivamente, para o cálculo.
- N é o conjunto de prováveis caminhos que a formiga tem para escolher.

O cálculo da atualização do feromônio é feito do seguinte modo:

$$\tau_{ij} = \rho(\tau_{ij}) + \Delta^{all} \quad (2.2)$$

Onde:

- τ_{ij} é o valor do feromônio no local pretendido.
- O símbolo ρ é o parâmetro que representa a taxa de evaporação do feromônio naquele local. Seu valor varia no intervalo de 0 a 1.
- $\Delta = E_{sol} / E_{opt}$. E_{sol} é a energia obtida a partir do caminho que a formiga percorreu. E_{opt} é a melhor energia conhecida da proteína. Se esta informação for desconhecida, usa-se a melhor energia encontrada até o momento da execução. No AS todas as formigas depositarão feromônio.
- Δ^{all} indica que todas as formigas depositam feromônio.

Na abordagem ACS, criada por [Dorigo e Gambardella \(1997\)](#) ocorre a atualização do feromônio local e globalmente. Isto significa que cada formiga, após concluir o processo de busca pela solução, depositará feromônio, sendo assim uma atualização local.

Após todas as formigas concluírem suas soluções, somente a melhor formiga da população encontrada até o momento, denominada *best-so-far*, irá novamente atualizar feromônio, sendo essa atualização chamada de atualização global.

No ACS o cálculo da probabilidade da formiga (P_{ij}) ir da posição i para a j é feito com algumas modificações. Primeiramente são definidas as variáveis q e q_0 . A variável q irá assumir um valor aleatório entre $[0,1]$ e q_0 é um valor de parâmetro dentro deste intervalo.

$$\begin{cases} \text{Se } q < q_0 & \text{então } P_{ij} = (\tau_{ij})^\alpha (\eta_{ij})^\beta, \\ \text{Senão} & , P_{ij} = (\tau_{ij})^\alpha (\eta_{ij})^\beta / \sum_{e \in N} (\tau_{ie})^\alpha (\eta_{ie})^\beta \end{cases}$$

Como nesta abordagem há uma atualização local e global do feromônio são usadas fórmulas distintas.

A atualização local do feromônio ocorre quando a formiga acaba de concluir seu caminho e gera uma possível solução para um problema. Isto é válido para todas as formigas da colônia. A fórmula desta atualização é a seguinte:

$$\tau_{ij} = (1 - \varphi)(\tau_{ij}) + \varphi\tau_0 \quad (2.3)$$

Onde:

- φ é um valor no intervalo $[0,1]$.
- τ_0 é o valor inicial do feromônio naquele local.
- τ_{ij} é o valor do feromônio no local.

A atualização global do feromônio no ACS é feita somente pela melhor formiga encontrada até o momento e sua fórmula é:

$$\tau_{ij} = (1 - \rho)(\tau_{ij}) + \rho\Delta^{melhor} \quad (2.4)$$

Onde:

- O símbolo ρ é o parâmetro e seu valor está no intervalo $[0,1]$.
- τ_{ij} é o valor do feromônio no local que se deseja ir.
- $\Delta = E_{sol} / E_{opt}$ é a energia obtida a partir do caminho que a formiga percorreu. E_{opt} é a melhor energia conhecida da cadeia proteica.

- Δ^{melhor} indica que o feromônio será atualizado pela formiga que encontrou o melhor caminho.

O MMAS (*Max-Min Ant System*), criado por [Stützle e Hoos \(1998\)](#), também tem atualização do feromônio por uma única formiga, que pode ser a melhor formiga da iteração ou a melhor formiga encontrada até o momento. O valor do feromônio não ultrapassa os limites $[T_{min}, T_{max}]$, que são valores mínimos e máximos que o feromônio pode assumir, a fim de evitar uma convergência prematura. As fórmulas da probabilidade e da atualização do feromônio para a abordagem do MMAS são as mesmas do AS, diferindo apenas em relação a formiga que atualiza o feromônio. Como esta abordagem foi utilizada neste projeto estas fórmulas estão mais detalhadas no Capítulo 5.

O algoritmo começa com uma quantidade inicial de feromônio. As formigas estão representadas pela letra K e variam até m . Todas as formigas constroem uma provável solução. Ao final de um certo número de interações é retornada a melhor solução encontrada.

A seguir um pseudocódigo do ACO é apresentado:

Algorithm 2 Pseudocódigo do ACO

- 1: Inicializar feromônio.
 - 2: **enquanto** número de iterações não atingido **faça**
 - 3: **para** $k \leftarrow 1$ **até** m **faça**
 - 4: Construir possível solução.
 - 5: Avaliar possível solução encontrada.
 - 6: Atualizar feromônio.
 - 7: **fim para**
 - 8: Guardar melhor solução até o momento.
 - 9: **fim enquanto**
 - 10: Retornar solução encontrada.
-

3 Problema de Predição de Estruturas de Proteína

3.1 Estruturas das Proteínas

As proteínas são moléculas formadas por ligações de aminoácidos, chamadas ligações peptídicas. São as proteínas que praticamente mantêm a vida e uma de suas funções vitais é o transporte de oxigênio para as células. Outra funcionalidade de uma proteína são as defesas naturais dos animais, por exemplo, que são os anticorpos, formados por proteínas. No corpo humano, as proteínas estão presentes na forma de enzimas, das células de defesa, exercendo função de suporte mecânico para células e tecidos, transportam nutrientes entre as células, entre outras funções.

Na natureza, existem vinte tipos diferentes de aminoácidos, que são compostos por carbono (C), hidrogênio (H), oxigênio (O) e nitrogênio (N). O grupo carboxílico é o COOH e o grupo amina é NH₂. A cadeia lateral R (ou radical) é uma sequência de elementos químicos ligados entre si, geralmente são hidrocarbonetos (compostos formados por ligações químicas entre carbono e hidrogênios). É esta cadeia que diferencia os aminoácidos uns dos outros.

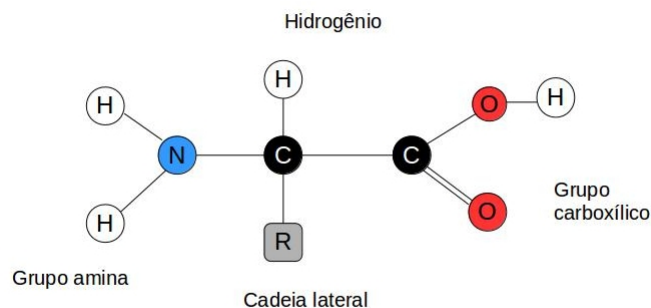


Figura 3 – Estrutura de um aminoácido.

Saber a estrutura de uma proteína é fundamental para descobrir suas funções, uma vez que a estrutura proteica tem uma relação direta com as suas funções. Conhecendo as funcionalidades destas moléculas torna-se mais fácil a manipulação pelos pesquisadores que as utilizam para fins biológicos. Muitas pesquisas biológicas utilizam as proteínas para encontrar, por exemplo, novos fármacos, estudar algumas doenças e vacinas. Portanto, encontrar a estrutura de uma proteína é um trabalho importantíssimo, porém este problema é muito difícil de ser resolvido, sendo caracterizado como um problema NP, em

termos computacionais.

Existem quatro tipos de estruturas de proteínas: estrutura primária, secundária, terciária e quaternária, e suas descrições seguem abaixo.

A **estrutura primária** de uma proteína é a mais simples, sendo definida como uma cadeia de aminoácidos, formada por ligações peptídicas. Estas ligações químicas ocorrem entre o grupo carboxílico de um aminoácido com o grupo amina de outro aminoácido. As ligações entre poucos aminoácidos são chamadas de peptídeo, e quando há muitos aminoácidos dá-se o nome de polipeptídeo. Pode-se dizer que uma proteína é um polipeptídeo.

A cadeia polipeptídica da estrutura primária pode enrolar sobre si mesma de várias maneiras, no meio em que esta se encontra, formando **estrutura secundária**. A cadeia pode enrolar-se em forma de espiral e, neste caso, recebe o nome de estrutura α -hélice. Na conformação α -hélice ocorre uma ligação química chamada ponte de hidrogênio, que liga o hidrogênio H do grupo amina de um aminoácido e o oxigênio O da carboxila do outro. Há a conformação β -folha, em que a cadeia de aminoácidos assume uma forma semelhante a uma folha dobrada em zigue-zague. Essas duas formas citadas são as principais da estrutura secundária.

A **estrutura terciária** se caracteriza pelo formato tridimensional da proteína, isto é, a forma gráfica que a proteína assume. Essa estrutura tridimensional está diretamente relacionada com a função que a proteína desempenha. Essa estrutura é definida pelo enovelamento da sua cadeia, sendo direcionado por diversos critérios, dentre eles a energia intramolecular, a energia da molécula com o meio que a envolve, a hidrofobicidade, dentre outros. Os aminoácidos hidrofóbicos (hidro=água; fobia=medo, aversão) da cadeia polipeptídica não interagem ao meio e se voltam para o centro da estrutura, enquanto que as partes hidrofílicas ou polares da cadeia se voltam para a superfície. Note que nesta estrutura pode haver diversas combinações de estruturas secundárias.

A **estrutura quaternária** ocorre quando a proteína apresenta mais de uma cadeia polipeptídica, e as estruturas terciárias se agrupam e interagem entre si formando moléculas maiores.

A Figura 4 apresenta os diferentes tipos de estruturas de proteínas.

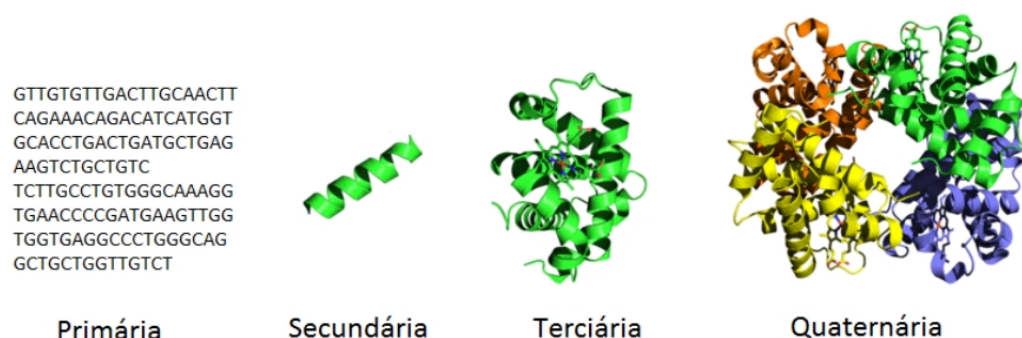


Figura 4 – Representação de estruturas das proteínas .

Fonte: (BRASIL, 2012)

3.2 Métodos para predição de estrutura

Existem alguns métodos para deduzir a estrutura tridimensional de proteínas. Os métodos convencionais para predição são a Cristalografia de Raio X e a Ressonância Nuclear Magnética. Esses métodos, porém, apresentam alto custo e gastam muito tempo. Os métodos computacionais surgiram como uma alternativa a estes métodos ainda limitados. As abordagens computacionais para predição de estrutura proteica são *ab initio*, *threading* e *homologia*.

A abordagem ***ab initio*** cria modelos de estruturas tridimensionais a partir da estrutura primária da proteína e das energias envolvidas no processo de enovelamento, ou seja, não se utilizam nenhum conhecimento prévio. A abordagem ***threading*** analisa a sequência de aminoácidos e verifica se existe uma sequência similar em um banco de dados de proteínas. Se houver uma sequência parecida, é um indicativo de que a estrutura poderá ser próxima a uma já existente. A **homologia** consiste em elaborar um modelo para a estrutura baseado em estruturas de proteínas homólogas, ou seja, aquelas que apresentam uma certa semelhança entre si e também pela classe familiar que já foram estudadas. Neste trabalho foi usada a abordagem ***ab initio*** .

Os modelos computacionais *ab initio* foram criados para que se tenha uma representação computacional da proteína com base nos princípios físicos e químicos dos aminoácidos, e não em estruturas já conhecidas nem proteínas homólogas. Nesta abordagem, as estruturas podem ser representadas utilizando os modelos: *full atom* , *lattice* e *off lattice* .

O modelo de representação *full atom* representa os átomos da proteína e os ângulos de torção referentes a eles. Esse modelo é mais completo e garante mais representatividade da proteína, sendo mais preciso, portanto. A desvantagem deste modelo é o alto custo computacional.

Modelos *lattice* são aqueles que usam redes reticuladas, ou malhas quadráticas para representar o espaço de busca por soluções do PSP. O problema em questão é modelado de modo que seja possível realizar movimentos no *lattice*, ou seja, na malha. A representação em modelos *lattice* discretiza o espaço de conformações e preserva as características importantes na computação de conformações de mínima energia. Um dos modelos *lattice* muito usado para o problema de predição de estruturas de proteínas é o modelo Hidrofóbico – Polar (HP).

Em contrapartida ao *full atom*, o modelo HP é bem mais simples, pois considera somente os aminoácidos da cadeia, sendo deste modo mais rápido comparado ao modelo *full atom*. Esse modelo foi escolhido neste projeto por ter uma implementação mais fácil e ser menos dispendioso computacionalmente. O conceito deste modelo será melhor detalhado a seguir.

O modelo *off-lattice* não usa malhas quadráticas. Sendo assim, a sequência de aminoácidos pode ser representada através de grafos. Hart e Istrail (1997) propõe transformar o grafo, que representa a proteína, em um conjunto de esferas de mesmo diâmetro. Cada vértice do grafo é substituído por uma esfera e cada aresta representará uma ligação tangente em uma conformação. O contato H-H é obtido quando há duas esferas hidrofóbicas tangentes.

3.3 Modelo HP para o problema PSP

O modelo Hidrofóbico-Polar (HP) é um modelo de representação de proteínas proposto pelos autores Lau e Dill (1989), e considerado um modelo baseado em redes (*lattice model*).

Esses modelos oferecem uma representação simplificada para o problema de predição de proteínas (PSP) e são bastante utilizados. Estes se caracterizam por usarem malhas reticuladas em duas dimensões (2D) ou em três dimensões (3D), onde os aminoácidos se posicionam nos vértices desta malha.

O modelo HP considera a hidrofobicidade e a polaridade dos aminoácidos para a representação computacional. A hidrofobicidade, no contexto biológico, define as moléculas que não se dissolvem nem interagem com a água, isto é, se repelem do meio aquoso, procurando se manter distantes dele. Na polaridade ocorre o oposto, ou seja, as moléculas possuem uma forte interação com a água.

Este modelo classifica os vinte aminoácidos existentes na natureza em grupos H ou P. No grupo H estão os aminoácidos hidrofóbicos e no grupo P, os aminoácidos hidrofílicos (polares). A Tabela 1 apresenta os aminoácidos conhecidos, suas abreviações e classificação quanto a hidrofobicidade e polaridade.

Aminoácido	Abreviação	Classe
Ácido Aspártico	Asp	Polar, hidrofílico
Ácido Glutâmico	Glu	Polar, hidrofílico
Alanina	Ala	Hidrofóbico
Arginina	Arg	Polar, hidrofílico
Asparagina	Asn	Polar, hidrofílico
Cisteína	Cys	Polar, hidrofílico
Fenilalanina	Phe	Hidrofóbico
Glicina	Gly	Polar, hidrofílico
Glutamina	Gln	Polar, hidrofílico
Histidina	His	Polar, hidrofílico
Isoleucina	Ile	Hidrofóbico
Leucina	Leu	Hidrofóbico
Lisina	Lys	Polar, hidrofílico
Metionina	Met	Hidrofóbico
Prolina	Pro	Hidrofóbico
Serina	Ser	Polar, hidrofílico
Tirosina	Tyr	Polar, hidrofílico
Treonina	Thr	Polar, hidrofílico
Triptofano	Trp	Hidrofóbico
Valina	Val	Hidrofóbico

Tabela 1 – Classificação dos aminoácidos quanto a polaridade e hidrofobicidade.

As interações entre aminoácidos hidrofóbicos são importantes para a formação da estrutura proteica, pois induzem com que as cadeias de aminoácidos se dobrem. Essas interações são muito bem representadas no modelo HP.

A Figura 5 apresenta uma cadeia de aminoácidos, de sequência PHHPPHHPHP, classificados em hidrofóbicos e polares dispostos no modelo HP, onde os símbolos H e P se posicionam nos vértices na malha 2D.

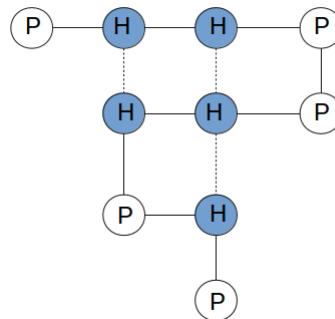


Figura 5 – Representação de estrutura de uma proteína no modelo HP.

Os nós azuis representam os aminoácidos classificados no grupo H e os brancos representam os aminoácidos polares do grupo P. A linha pontilhada representa a interação entre os aminoácidos hidrofóbicos vizinhos não conectados.

Sabe-se que a estrutura terciária começa a se formar pelo dobramento da sequência de aminoácidos. O dobramento ocorre devido as ligações entre aqueles resíduos que se encontram a uma certa distância, isto é, aqueles que são vizinhos não conectados. Por isso, a energia livre de conformação é obtida pelas interações H-H de aminoácidos não conectados.

Essa energia é inversamente proporcional à quantidade de interações H-H, uma vez que a função objetivo do PSP busca a menor energia livre possível. A energia de conformação é dada pela seguinte fórmula:

$$E = \beta_{i,j} \sum \delta(r_i, r_j) \quad (3.1)$$

Onde:

- β assumirá 1 se os aminoácidos forem do tipo H e 0, caso contrário.
- A função δ assume 1 se os aminoácidos r_i e r_j são vizinhos não conectados, e 0, caso contrário.

4 Trabalhos Correlatos

O Algoritmo Evolutivo (AE) e a Otimização por Colônia de formigas (ACO) foram sugeridos inicialmente por John Holland e Marco Dorigo, respectivamente, como métodos para solucionar problemas de grandes complexidades computacionais. Com base nesta teoria, outros pesquisadores começaram a aplicar estes métodos a problemas reais, como o problema de predição de estruturas de proteínas, descrito no Capítulo 3. Este Capítulo 4 apresenta alguns trabalhos com estas técnicas de otimização que se relacionam com o tema tratado neste trabalho.

Há vários estudos nesta área de pesquisa, tendo como um dos pioneiros o artigo de [Khimasia e Coveney \(1997\)](#), que apresenta o algoritmo genético convencional como uma boa estratégia de solução para o problema PSP e algumas orientações para melhorar seu desempenho, como uma busca pelos parâmetros para determinar os melhores valores para o AG e teste de várias funções de avaliação da energia. Os autores utilizam o modelo HP em malha 3D.

Diversos trabalhos similares foram desenvolvidos em seguida, como o trabalho de Custódio ([CUSTODIO; BARBOSA; DARDENNE, 2004](#)) que realizou um estudo sobre a busca *ab initio* de estruturas de proteínas, usando modelo de representação HP em malha 3D e algoritmos genéticos. Esses autores apresentam uma modificação na avaliação do resultado, baseando-se no fato de que os aminoácidos H tendem a não ficar em contato com o meio os quais têm aversão. Também é usado um *island's algorithm* ([WHITLEY, 1994](#)) para melhorar o desempenho do AG. A ideia deste algoritmo é executar várias vezes o AG com poucos indivíduos e em poucas gerações e utilizar as melhores estruturas encontradas para criar a população inicial de um novo AG.

Seguindo a ordem cronológica, surgiram outros trabalhos que utilizaram outros modelos de representação, como o trabalho de [Cutello, Narzisi e Nicosia \(2006\)](#) que usa para a representação de proteínas o *modelo full atom*. Neste contexto, têm-se outros trabalhos similares ([BRASIL, 2012](#)), ([BONETTI, 2015](#)), em que se pode citar a tese de doutorado de [Brasil \(2012\)](#) que apresenta uma abordagem do PSP *ab initio* no modelo *full atom*, onde foram desenvolvidos algoritmos evolutivos de múltiplos objetivos. Diferente do modelo HP, considera todos os átomos da molécula e os ângulos de torção referentes a eles. Esta abordagem costuma apresentar melhores resultados por ser mais preciso, no entanto, é muito mais lento.

Os estudos realizados por [Johnson e Katikireddy \(2006\)](#) mostram o algoritmo evolutivo para tratar do problema de PSP com o modelo HP, com o diferencial de oferecer *backtracking* em sua implementação, o que possibilita maior otimização no processo de

encontrar a solução. Neste trabalho, é usado o modelo de representação HP em malha 3D. Nesta monografia de conclusão de curso, utilizou-se a técnica de *backtracking* inspirada neste artigo.

Com o modelo HP, pode-se citar a dissertação de mestrado de [Gabriel \(2010\)](#), que menciona o problema PSP e como ele pode ser solucionado pelo AG com formulações multiobjetivos. As estruturas geradas pelo AG são avaliadas de forma mais eficiente, baseado no número de contatos hidrofóbicos com a distância entre os resíduos.

Um importante trabalho com o modelo HP é o artigo do [Huang e He \(2010\)](#). Esse artigo apresenta o AE para PSP com o modelo de representação HP e malha bidimensional. Huang propõe uma melhoria no AE com base nas estruturas secundárias no modelo HP. Resumidamente, a ideia desta melhoria é descobrir uma subcadeia formada por aminoácidos hidrofóbicos, de modo que esta subcadeia pode vir a formar estruturas secundárias, como β -folha e α -hélice. Uma vez descobertas, essas subcadeias só se podem se tornar β -folha ou α -hélice, não podendo sofrer outro tipo de mutação nas próximas gerações. Essas melhorias mostram que o AE ficou mais eficiente em relação a busca da solução ótima quando comparado ao AE ([UNGER; MOULT, 1993](#)) e outros métodos, como o algoritmo de Monte Carlo ([LIANG; WONG, 2001](#)). A solução ótima é baseada no cálculo da energia de conformação dada pelo modelo HP, explicado na Seção 4.3.

Os AEs também podem apresentar ótimas soluções quando são hibridizados com outros métodos. O estudo de [Lin e Su \(2011\)](#) é um exemplo de AE híbrido com PSO (*Particle Swarm Optimization*) utilizando malha 3D e modelo HP. Os resultados deste modelo híbrido é comparado com o AE padrão e outros métodos, e se mostram bem eficientes.

Além dos trabalhos citados sobre AE aplicados ao problema PSP, se destacam aqueles que abordam o ACO para resolver este problema, visto que este algoritmo é proposto como um bom solucionador para problemas complexos.

[Shmygelska, Hernández e Hoos \(2002\)](#) foi a primeira aplicação do ACO para solucionar o PSP. O artigo apresenta a resolução do PSP com o ACO usando o modelo de representação proteico HP nas malhas 2D. Após a publicação desse artigo, os autores continuaram as pesquisas neste tema, fazendo melhorias com base no primeiro estudo. O artigo de [Shmygelska e Hoos \(2003\)](#) mostra esta melhoria, que consiste em um tipo de movimento de longo alcance. Resumidamente isto ocorre da seguinte maneira: na busca local, um dos aminoácidos originais do movimento é escolhido aleatoriamente e têm sua direção modificada também aleatoriamente. Os aminoácidos seguintes são alocados de acordo com a posição do aminoácido anterior e seguindo os cálculos probabilísticos do ACO, que a formiga realiza para decidir qual será sua próxima posição.

No artigo de [Shmygelska e Hoos \(2005\)](#) é apresentado a resolução do PSP com

o ACO usando o modelo de representação proteico HP nas malhas 2D e 3D, seguindo a melhoria feita no artigo anterior. Este artigo se relaciona com este projeto de TCC pelo fato de usar malhas bidimensionais, modelo HP e apresentar um comparativo entre vários métodos usados para resolver o PSP, dentre eles o AE.

Outro pesquisador que fez estudos sobre como o ACO com bons resultados para o problema PSP foi [Chu, Till e Zomaya \(2005\)](#) . O autor sugere o ACO com o paradigma de programação distribuída para se alcançar estruturas. Ele também aplica o modelo de representação HP em malhas 2D e 3D.

Em trabalhos mais recentes como em [Citrolo e Mauri \(2013\)](#), é apresentado um método híbrido com o método Monte Carlo via Cadeia de Markov (MCMC) e o ACO. Esses pesquisadores também usaram o modelo HP e malha reticulada tridimensional. Os resultados desta nova heurística híbrida são comparados com o ACO e com o método *Simulated Annealing (SA)* , onde mostram resultados satisfatórios.

5 Implementação do Trabalho

Os algoritmos ACO e AE foram implementados em linguagem C, no sistema operacional Linux e em um computador processador intel *i7* com 4G (giga) de memória.

Os mesmos algoritmos foram implementados em meu trabalho de Iniciação Científica aplicados ao problema do caixeiro viajante (DIAS; BRASIL, 2015). Neste trabalho, os algoritmos foram reimplementados, sendo adaptados para o PSP (DIAS; BRASIL, 2016).

As figuras referentes a estrutura das proteínas foram geradas automaticamente no final da execução do AE e do ACO, por meio do uso da biblioteca *graphics.h*, da linguagem C.

5.1 Descrição do método ACO para o PSP

O ACO é inicializado com a criação de uma população de formigas. Em seguida, cada formiga constrói um caminho na malha, o qual representa uma estrutura da proteína, ou seja, uma possível solução para o PSP. Elas repetem este processo de construção em várias iterações. Ao final de cada solução construída a formiga atualiza o feromônio. Na natureza, o feromônio é uma substância que auxilia a colônia a encontrar e reconhecer caminhos. O ACO mimetiza este processo natural, e geralmente o representa em uma matriz.

Neste projeto foi utilizada a abordagem MMAS, descrita no Capítulo 2. Esta abordagem foi escolhida com o propósito de se aproveitar as melhores formigas para a atualização do feromônio, descartando as soluções ruins. O MMAS também se mostra mais eficiente quando comparado às demais abordagens.

5.1.1 Representação computacional da formiga

Inicialmente, cria-se uma população de formigas, e logo após as formigas constroem um caminho pela malha 2D, que representa uma solução para o PSP. Este caminho construído pela formiga pode ser compreendido como um vetor cujas posições possuem os movimentos que a formiga realizou. Na implementação a formiga é representada por uma estrutura que contém um vetor para movimentos, uma variável para armazenar a energia da possível solução e vetores para guardar as coordenadas cartesianas dos pontos do caminho. O conjunto de coordenadas são do tipo $[(x_i, y_i), \dots, (x_n, y_n)]$, onde $i = 1, 2, 3, \dots, n$, onde n é o tamanho da proteína, que são basicamente os pontos da malha por onde a formiga passou convertidos em valores de um plano cartesiano. Os movimentos que a formiga realiza serão detalhados a seguir.

5.1.2 Movimentação da formiga

No ACO para o PSP, a formiga é o agente responsável pela descoberta de uma possível estrutura para uma dada proteína. No modelo de representação HP, a formiga percorre alguns pontos da malha para formar a estrutura.

Considerando uma proteína qualquer no modelo HP de sequência HPHHPH, de acordo com a Figura 6, o aminoácido H destacado em vermelho indica que este foi o ponto de partida da formiga. As setas em cinza indicam quais foram os locais por onde ela passou. Ao passar por um local a formiga “deixa” um aminoácido e em seguida escolhe qual o próximo local (indicados pelas setas azuis) para o aminoácido seguinte da cadeia.

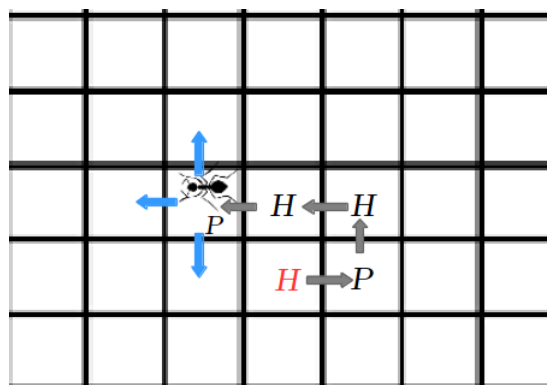


Figura 6 – Deslocamento da formiga.

Na implementação para este trabalho, o ponto de partida é o centro da malha, pois assim possibilita quatro movimentos para a formiga. Em todos os pontos do caminho da formiga são colocados os aminoácidos da cadeia. Por exemplo, no ponto inicial do caminho deve ser colocado o primeiro aminoácido da cadeia proteica, no segundo ponto em que a formiga passou está o segundo aminoácido da cadeia, e assim por diante. A orientação inicial é escolhida de forma aleatória, dentro das possibilidades dos movimentos. As escolhas iniciais na orientação aumenta a possibilidade de novos caminhos.

Na Figura 7, as setas indicam as direções de movimentos que a formiga pode ter na malha. As setas S1 e S3 mostram um movimento na horizontal, porém em S1, a formiga seguirá para a direita enquanto em S3 ela seguirá para a esquerda na malha. S2 e S4 indicam movimentos na vertical. A formiga pode ter um movimento de ir para cima em S2 ou ir para baixo em S4.

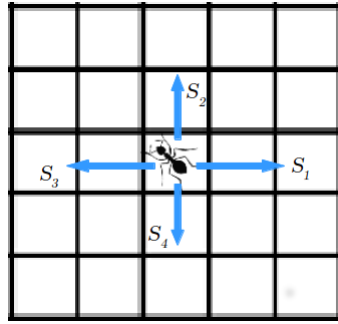


Figura 7 – Sentidos de movimentos da formiga.

Para percorrer a malha a formiga pode executar três movimentos, os quais são: ir para frente (F), virar à direita (R) ou virar a esquerda (L). Ao representar esses movimentos no vetor de movimentos da formigas, são dadas enumerações a estes movimentos. O movimento F é representado pelo número 0, o movimento R pelo número 1 e o movimento L é representado pelo número 2. Porém, ao escolher um desses movimentos, a orientação na malha é alterada. Por exemplo, a formiga estava no ponto de origem sentido horizontal à direita e escolheu o movimento virar à direita (R), logo após realizar esse movimento a orientação da formiga passou a ser no sentido vertical para baixo. Essa orientação também precisa ser conhecida na implementação do problema.

As Figuras 8 e 9 mostram os movimentos que a formiga pode realizar quando sua orientação se encontra na vertical para cima e seu deslocamento na horizontal.

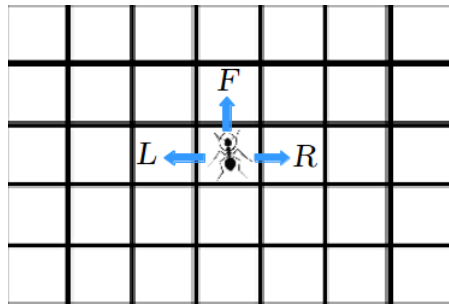


Figura 8 – Formiga com orientação na vertical.

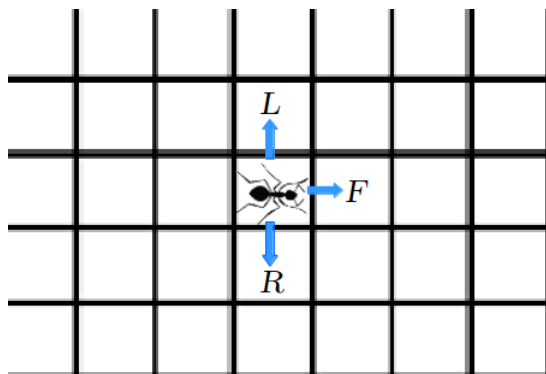


Figura 9 – Formiga com orientação na horizontal.

Definidos o ponto inicial e a orientação, a formiga escolhe a próxima posição com base em um cálculo probabilístico com as movimentações F, R e L. São analisados os cálculos de probabilidades de ir para F, R e L. A maior probabilidade é escolhida e a formiga caminha para aquele local correspondente.

Na malha 2D não é permitido que dois aminoácidos ocupem o mesmo lugar ao mesmo tempo. Quando isto ocorre há uma colisão. Para prevenir essa colisão, a formiga verifica se há a possibilidade de ir para pelo menos um dos três sítios possíveis, de acordo com os movimentos (F,R,L). Caso não seja possível ir para nenhuma das três posições, estando estas ocupadas, a formiga inicia um processo de *backtracking*. Este processo consiste em marcar a posição atual da formiga como posição inválida, de modo que a formiga não volte para ela. Logo após, a formiga volta para a posição anterior a sua posição atual. O *backtracking* pode fazer a formiga voltar quantas vezes forem necessárias, até encontrar uma posição em que não dê possibilidades de ocorrer colisões.

5.1.3 Cálculo da probabilidade

O cálculo de probabilidade da formiga ir para uma nova posição é efetuado para cada um dos três movimentos (F, R, L), quando as posições indicadas por estes movimentos se encontram livres. Deste modo, é feito um cálculo que utiliza o feromônio daquele local e uma informação heurística, explicada a seguir.

Inicialmente, o valor da informação heurística (η) é 1, sendo calculado da seguinte forma: considere que a formiga está tentando ir para a posição livre F_{pos} indicada pelo movimento F. Se o aminoácido a ser colocado neste ponto for do tipo H, ela olhará para todos os vizinhos de F_{pos} (locais indicados pelas letras c , d e e , mostradas na figura abaixo). A cada vizinho H será somado 1 ao valor da heurística ($\eta = \eta + 1$). Assim, seu maior valor será 4, para a posição F_{pos} , uma vez que F_{pos} possui três vizinhos. Contudo, se o aminoácido a ser colocado for P, a heurística (η) permanece com seu valor inicial ($\eta = 1$) e o cálculo da probabilidade é feito normalmente. Este processo se repete para as demais posições indicadas pelos movimentos R e L.

A Figura 10 mostra quais as vizinhanças que devem ser verificadas para calcular o valor da heurística η . Para calcular a heurística na posição indicada pelo movimento F, por exemplo, é necessário verificar em seus pontos vizinhos c , d , e e a existência de algum aminoácido H. Se essas posições estiverem vazias ou ocupadas por aminoácidos P, η não terá seu valor acrescido.

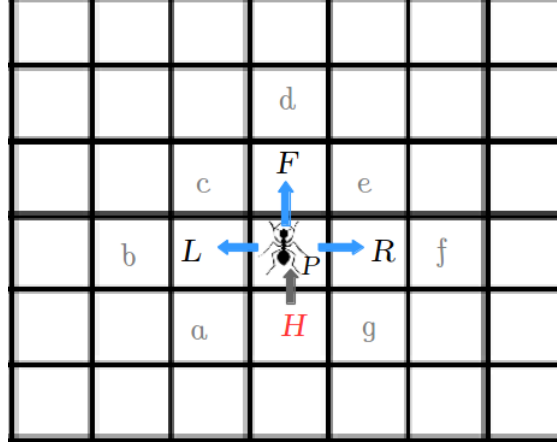


Figura 10 – Cálculo do valor heurístico.

A probabilidade de escolher a próxima posição é calculada pela seguinte fórmula:

$$P_{ij} = (\tau_{ij})^\alpha (\eta_{ij})^\beta / \sum_{e \in R, F, L} (\tau_{ie})^\alpha (\eta_{ie})^\beta \quad (5.1)$$

- As letras i e j significam, respectivamente o ponto em que a formiga está e o ponto que a formiga pode ir.
- Os expoentes α e β dão a importância que o feromônio (τ) e a informação heurística (η) terão.

5.1.4 Matriz de feromônios

Assim que a formiga conclui seu caminho pela malha e uma possível estrutura é encontrada, ela atualiza seu feromônio. O feromônio é uma substância que permanece por algum tempo no caminho em que a formiga percorreu. Para solucionar o problema PSP, a formiga percorre uma malha bidimensional, que é representada por uma matriz. A matriz de feromônios corresponderá a malha bidimensional, onde cada posição da matriz de feromônio é associado à posição corresponde da malha. Assim, as posições que a formiga percorreu na malha serão as mesmas posições em que será depositado feromônio na matriz de feromônios. Inicialmente o valor do feromônio é 1 para todas as posições.

5.1.5 Atualização do feromônio

A formiga, ao terminar seu passeio na malha, deposita uma quantidade de feromônio nesse caminho. A quantidade de feromônio pode ficar alta quando muitas formigas passam pelo mesmo caminho, porém quando o percurso não representa uma boa solução as demais formigas não seguirão aquele caminho e, por isso, a quantidade de feromônio presente ali sofrerá uma evaporação.

Essa implementação é a variação MMAS do ACO, onde somente uma formiga deposita feromônio, podendo ser a melhor encontrada até o momento ou a melhor da interação. Sendo assim, escolheu-se a melhor formiga encontrada até o momento.

O depósito de feromônio considera a evaporação e é dado pela seguinte fórmula:

$$\tau_{ij} = (\rho)(\tau_{ij}) + \rho\Delta^{melhor} \quad (5.2)$$

Onde:

- O símbolo ρ é o parâmetro que representa a taxa de evaporação do feromônio naquele local. Seu valor varia no intervalo de 0 a 1.
- $\Delta = E_{sol}/E_{opt}$ é a energia obtida a partir do caminho que a formiga percorreu. E_{opt} é a melhor energia conhecida da cadeia proteica até o momento da execução.
- Δ^{melhor} indica que o feromônio será atualizado pela formiga que encontrou o melhor caminho.

5.2 Descrição do método AE para o PSP

De modo semelhante ao ACO, a abordagem do AE para tratar o PSP se inicia com a criação de uma população de indivíduos. Inicialmente os indivíduos são diferentes entre si, tal como ocorre na natureza e evoluem para atingir a solução ótima.

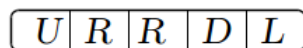
5.2.1 Representação computacional do indivíduo

Computacionalmente, o indivíduo é um vetor e suas posições são denominadas genes no AE. Uma população, que é um conjunto de indivíduos, geralmente é uma matriz ou um vetor do tipo do indivíduo.

Considerando o AE para resolver o PSP, o indivíduo é uma sequência de movimentos que a cadeia proteica pode ter, onde o conteúdo de cada gene especifica um movimento para cada um dos aminoácidos, com exceção do primeiro aminoácido da cadeia. Os movimentos que a cadeia pode fazer na malha são: ir à direita (R), ir à esquerda (L), ir para cima (U) e ir para baixo (D). O funcionamento destes movimentos na malha serão explicados mais adiante neste capítulo.

A Figura 11 representa um indivíduo (vetor) com seus genes contendo os movimentos a serem realizados para formar a estrutura.

Cromossomo



As letras U, R, L, D representam os movimentos a serem realizados.

Para simplificar, os movimentos são enumerados da seguinte forma:
R = 0, L = 1, U = 2, D = 3.

O cromossomo assumirá, então a seguinte forma:

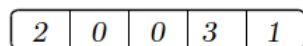


Figura 11 – Representação dos genes.

5.2.2 Formação da estrutura bidimensional da proteína no AE

A formação da estrutura bidimensional da proteína é criada pela movimentação dos aminoácidos da cadeia proteica na malha. Isto significa que a proteína assume direções na malha bidimensional para se mover. São quatro direções de movimentos possíveis: ir

à direita (R), ir à esquerda (L), ir para cima (U) e ir para baixo (D). Estes movimentos são os mesmos para todos os pontos da malha 2D.

De acordo com a Figura 12, considere o círculo na malha como sendo o primeiro aminoácido da cadeia proteica. O segundo aminoácido será alocado em uma das quatro posições indicadas pelas setas. A posição correta será definida pela informação que está no gene do indivíduo. Os demais aminoácidos seguem a mesma lógica.

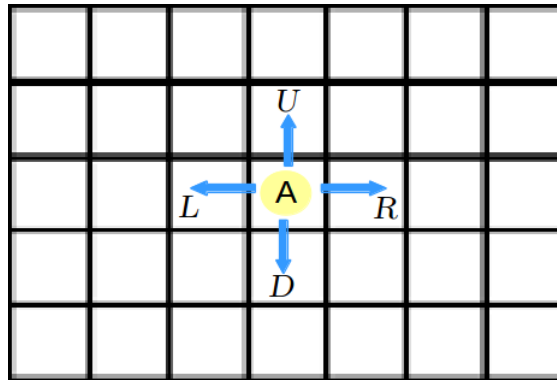


Figura 12 – Alocação dos aminoácidos pelo AE.

O processo de predição de estrutura começa com o primeiro aminoácido da cadeia colocado em um ponto na malha. Nessa implementação, foi escolhido o ponto central da malha, pois permite maior movimentação ao restante da cadeia na malha. O segundo aminoácido é colocado em uma das quatro posições (R, L, U, D), indicadas pelo gene. O primeiro gene do indivíduo informa qual a posição do segundo aminoácido em relação ao anterior, e assim por diante. Portanto o tamanho do indivíduo é sempre $n-1$, onde n é o tamanho da cadeia proteica.

A Figura 13 mostra como a sequência de aminoácidos se dobra na malha de acordo com a informação do cromossomo.

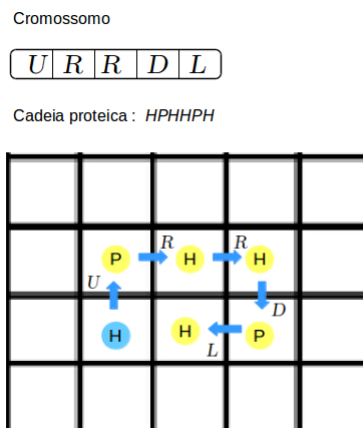


Figura 13 – Dobramento da proteína feita pelo AE.

No entanto, os indivíduos podem representar soluções que apresentam *colisões*, quando dois aminoácidos ocupam o mesmo lugar na malha. Nesta abordagem os indivíduos com essas características não são descartados. Sendo assim, estes indivíduos passam por um processo de correção desse problema.

O tratamento da colisão ocorre no momento em que os aminoácidos estão sendo dispostos na malha, de acordo com os genes do indivíduo. Verifica-se se o lugar em que o aminoácido ocupará está ocupado por outro ou vazio. Se este lugar já estiver ocupado, o gene que o indica sofrerá uma mutação no valor, de modo que indique para um lugar viável. Este processo pode ser executado quantas vezes forem necessárias, até o momento em que a solução proposta pelo indivíduo deixa de apresentar colisões.

5.2.3 Operações de *crossover*, mutação e seleção dos pais

Inicialmente, é preciso selecionar os indivíduos pais para a operação de *crossover*. Existem alguns métodos para a escolha dos pais, como o método da roleta, torneio de dois, torneio de três ou mesmo a escolha aleatória entre todos da população. No método da roleta os melhores pais têm maior probabilidade de serem escolhidos. O torneio de dois escolhe dois indivíduos da população, aleatoriamente ou pela roleta, e seleciona o melhor. De modo semelhante ao torneio de dois, o torneio de três seleciona o melhor indivíduo entre três indivíduos escolhidos. Para este projeto foi utilizado a seleção aleatória dos pais entre todos os indivíduos da população.

Após a seleção dos pais, os mesmos realizam o *crossover*. Neste trabalho foram empregados dois tipo de *crossovers*, os quais são os de 1-ponto e 2-pontos.

Nas Figuras 14 e 15 estão representados os tipos de *crossover* usados. As linhas tracejadas indicam onde houve o ponto de corte nos cromossomos.

Crossover 1-ponto

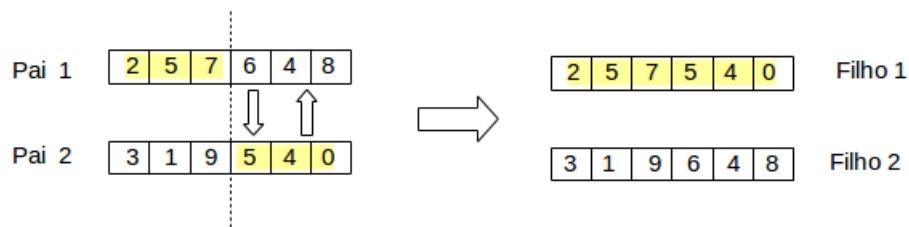
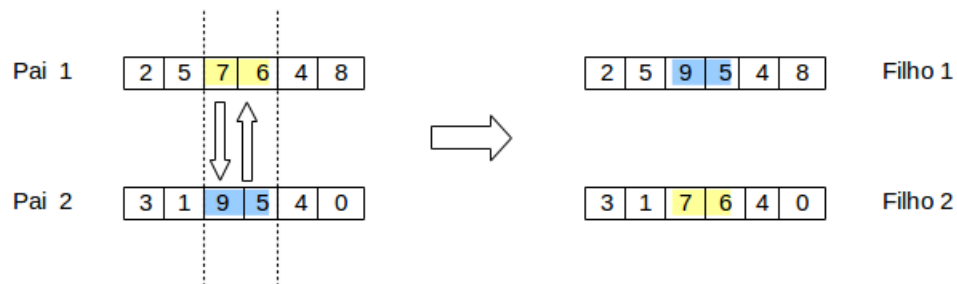


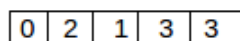
Figura 14 – *Crossover* 1-ponto.

Crossover 2-pontosFigura 15 – *Crossover* 2-pontos.

Os dois tipos de *crossover* existentes nessa implementação foram executados de acordo com a ocorrência de gerações executadas. Por exemplo, nas 10% das primeiras gerações somente o *crossover* 1-ponto foi executado. Após estas gerações, foi dada uma percentagem de 50% de ocorrer o *crossover* 1-ponto ou o *crossover* 2-pontos.

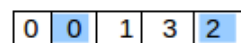
A mutação neste trabalho foi baseada na mutação apresentada por (HUANG, 2010). Esta operação ocorre sobre um ou mais indivíduos da população, escolhidos aleatoriamente. O processo da mutação ocorre pela escolha aleatória de um número (s) entre 1 e $n/2$, onde n é tamanho do indivíduo. Este número aleatório (s) representa a quantidade de genes que sofrerá mutação. Não é satisfatório o número zero fazer parte deste intervalo, pois se ele fosse sorteado não haveria mutação, contrariando o próprio operador.

Considerando s um número aleatório entre 1 e $n/2$, onde n é o tamanho do indivíduo e supondo que $n=5$ e o resultado da escolha aleatória seja $s=2$, significa que dois genes do indivíduo sofrerão mutação. Estes genes podem estar em qualquer lugar no cromossomo, escolhidos aleatoriamente. Na Figura 16 os genes destacados significam que esses foram escolhidos aleatoriamente e sofreram mutação.

MutaçãoCromossomo com tamanho $n = 5$ 

Sorteia-se um número aleatório entre 1 e $n/2$.

Supondo que o número sorteado seja 2.



Escolhe-se dois genes aleatórios do cromossomo, os quais sofrerão mutação.

Figura 16 – Implementação da mutação.

Para evitar a perda dos melhores indivíduos ao longo das gerações, usa-se o elitismo. O elitismo consiste em identificar os melhores indivíduos de uma geração e transferir os mesmos para a geração seguinte. Nesta implementação a quantidade de indivíduos mantidos para a próxima geração é equivalente à metade da população.

6 Resultados e Discussões

Neste capítulo serão apresentados os resultados comparativos entre os dois métodos (ACO e AE) para resolver o problema PSP. Ambos utilizaram o mesmo conjunto de proteínas advindas do artigo de [Shmygelska e Hoos \(2005\)](#).

O ACO usa os parâmetros α para expressar a importância do feromônio e β para indicar a importância da informação heurística. Também é utilizado um parâmetro para a atualização do feromônio, sendo representado pela letra grega ρ . Os resultados foram obtidos com $\alpha = 1$, $\beta = 1$ e $\rho = 0,8$. A abordagem MMAS, utilizada nesta implementação, mantém o valor do feromônio em um intervalo $[T_{min}, T_{max}]$, onde $T_{min} = 1$ e $T_{max} = 2(\rho + 1)$. A fórmula para T_{max} foi uma alteração baseada no artigo de [Stützle e Hoos \(2000\)](#), onde este utiliza o parâmetro ρ para a atualização do feromônio para calcular os limites.

Foram utilizados no AE dois tipos de *crossover*, o *crossover* de 1-ponto e *crossover* de 2-pontos, e um tipo de mutação baseada em ([HUANG; HE, 2010](#)). Essas operações foram utilizadas no AE do seguinte modo: sendo N o número de gerações, da geração 1 até $0.1 \cdot N$ (ou seja, 10% de N) a escolha entre os dois tipos de *crossover* ficou em 50% para o *crossover* de 1-ponto e 50% *crossover* de 2-pontos. A probabilidade de ocorrer a mutação na população ficou em 30%. Após as $0.1 \cdot N$ gerações, até a conclusão de 70% das gerações, a probabilidade de escolha ficou em 70% para o *crossover* de 1-ponto e 30% para o outro tipo. Nesta etapa a probabilidade da mutação foi alterada para 40%. Após a conclusão de 70% do número de gerações, somente o *crossover* de 1-ponto foi executado e a taxa da mutação mudou para 50%. A escolha dos pais foi feita de forma aleatória entre todos os indivíduos da população para as proteínas de 20 a 36. Para a proteína maior, de 48 aminoácido, foi utilizado o torneio de três para a seleção.

A tabela da Figura 17 mostra as sequências das proteínas utilizadas nas execuções do ACO e do AE, no modelo HP, seus tamanhos e suas energias ótimas.

Foram feitas 10 execuções para cada configuração, onde foram variados os números de indivíduos, formigas, gerações e iterações, descritas nas tabelas das seções a seguir:

Tamanho da proteína	Energia	Proteína
20	-9	HPHPPHHPHPPHHPHPH
24	-9	HHPHPHPHPHPHPHPHPHH
25	-8	PPHPHHPPPHHPPPHHPPHH
36	-14	PPHHPPHHPPPPHHHHHHHPH HPPHHPPHP
48	-23	PPHPHHPPHHPPPPHHHHHHHH HHPPPPPHHPPHHPPHHHHHH

Figura 17 – Proteínas utilizadas para obter os resultados.

6.1 Resultados com a proteína de 20 aminoácidos

A sequência avaliada foi HPHPPHHHPHPHPHPPHPH. A energia ótima desta proteína é -9.

Tabela 2 – Resultados das execuções do ACO.

NumF	N	E_{enc}	Me_{ACO}	Var_e	T_{ACO}
100	100	-9	-8.7	0.21	284.00
500	500	-9	-9.0	0.0	4408.60
10	10000	-9	-9.0	0.0	1745.59
20	10000	-9	-9.0	0.0	5763.20
30	10000	-9	-9.0	0.0	8724.70
40	10000	-9	-9.0	0.0	11680.50
50	10000	-9	-9.0	0.0	14506.09
60	10000	-9	-9.0	0.0	17576.09
70	10000	-9	-9.0	0.0	20530.30
80	10000	-9	-9.0	0.0	23606.80
90	10000	-9	-9.0	0.0	26448.69
100	10000	-9	-9.0	0.0	29323.50

Tabela 3 – Resultados das execuções do AE.

NumInd	N	E_{enc}	Me_{AE}	Var_e	T_{AE}
100	100	-8	-7.2	0.36	151.3
500	500	-9	-8.1	0.29	3756.30
10	10000	-9	-7.9	0.49	1689.80
20	10000	-9	-7.9	0.69	4984.29
30	10000	-9	-7.8	0.56	8599.79
40	10000	-9	-7.7	0.81	11401.00
50	10000	-9	-8.0	0.40	13840.90
60	10000	-9	-7.9	0.49	16575.80
70	10000	-9	-7.7	0.41	20501.80
80	10000	-9	-7.7	0.61	20017.690
90	10000	-9	-8.2	0.56	23842.90
100	10000	-9	-8.2	0.36	24475.30

Onde:

- NumInd é o número de indivíduos no AE.
- NumF é o número de formigas no ACO.
- N é o número de iterações/gerações.
- E_{enc} é a melhor energia encontrada em 10 execuções.
- Me_{ACO} e Me_{AE} são as médias aritméticas das energias obtidas pelo ACO e AE, respectivamente, em 10 execuções.
- Var_e é a variância populacional da média das energias de 10 execuções.
- T_{ACO} e T_{AE} é a média aritmética do tempo (em milissegundos) de execução do algoritmo em 10 execuções, do ACO e AE, respectivamente.

De acordo com os resultados apresentados nas Tabelas 2 e 3, o método ACO teve melhor desempenho em relação ao número de respostas ótimas encontradas, pois as suas médias (Me_{ACO}) foram melhores que as médias do AE (Me_{AE}). Embora o AE tenha obtido a energia ótima em 91% das execuções, pode-se observar pelos valores da variância (Var_e) que no ACO são menores que no AE, em todas as configurações, mostrando que o ACO foi mais eficiente ao encontrar a solução ótima. Porém, do ponto de vista computacional, o tempo no AE foi um pouco menor que o ACO.

A Figura 18 foi retirada do artigo de [Huang e He \(2010\)](#) e as Figuras 19 e 20 foram os resultados das estruturas obtidas pelos algoritmos ACO e AE, respectivamente. Pode-se observar que, para esta proteína, as estruturas foram semelhantes em geral, diferenciando em suas rotações. Visualmente pode-se destacar também o ACO, analisando a Figura 19(b), que equivale a estrutura ótima da Figura 18.

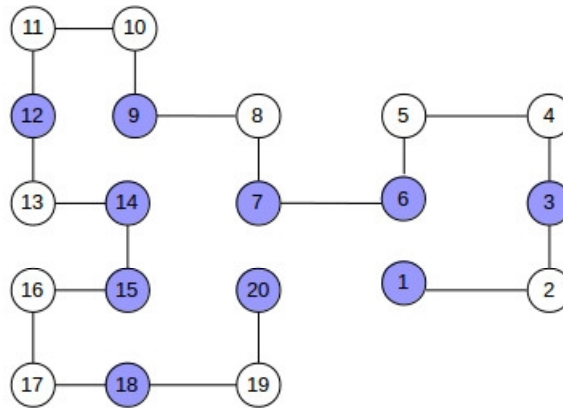


Figura 18 – Estrutura ótima da proteína de 20 aminoácidos.

Fonte: (HUANG; HE, 2010)

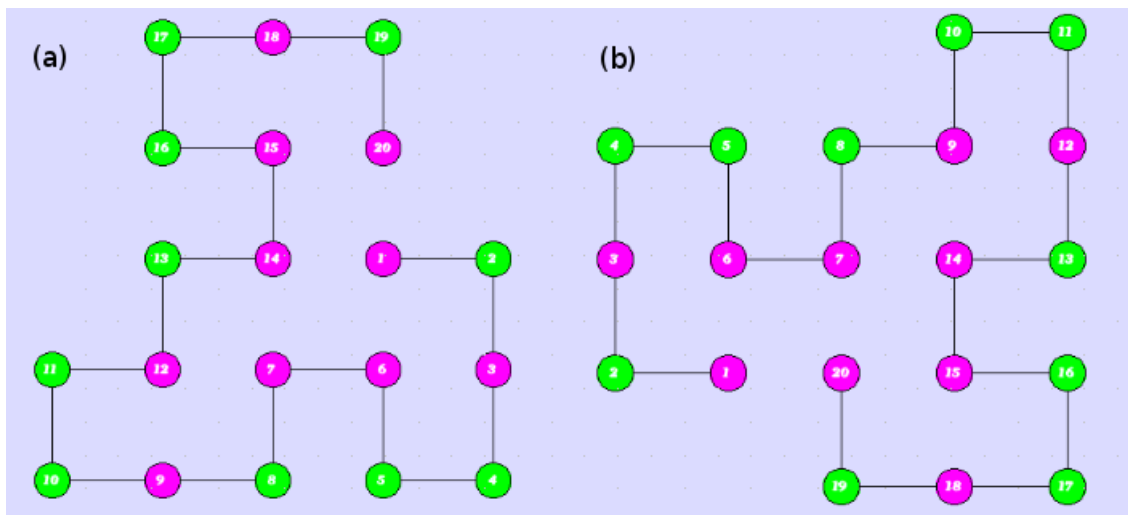


Figura 19 – Estruturas ótimas encontradas pelo ACO.

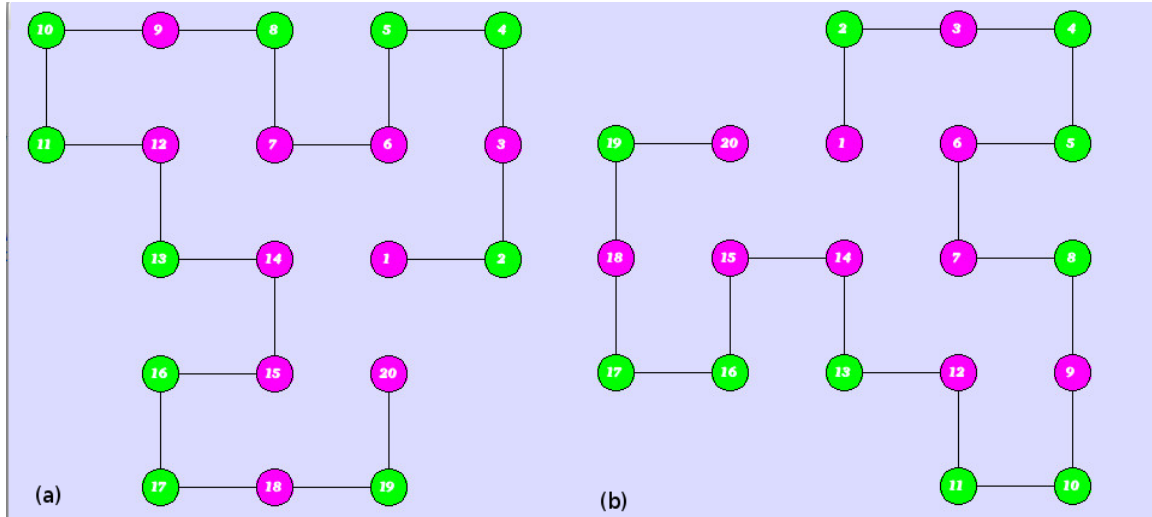


Figura 20 – Estruturas ótimas encontradas pelo AE.

Na Figura 18 retirada do artigo de referência, os círculos brancos representam os aminoácidos polares e os azuis representam os aminoácidos hidrofóbicos. Nas estruturas obtidas pelos algoritmos (Figuras 19 e 20), os círculos em verde representam os aminoácidos polares (P) e os em rosa representam os aminoácidos hidrofóbicos (H).

6.2 Resultados com a proteína de 24 aminoácidos

A proteína de sequência HHPHPPHPPHPPHPPHPPHPPHH foi avaliada e sua energia ótima é -9. As Tabelas 4 e 5 apresentam os resultados das execuções do ACO e do AE, respectivamente.

Tabela 4 – Resultados das execuções do ACO.

NumF	N	E_{enc}	Me_{ACO}	Var_e	T_{ACO}
100	100	-9	-8.1	0.09	334.70
500	500	-9	-8.9	0.09	7898.60
10	10000	-9	-8.7	0.21	3231.10
20	10000	-9	-9.0	0.0	6220.89
30	10000	-9	-9.0	0.0	9696.09
40	10000	-9	-9.0	0.0	12494.50
50	10000	-9	-9.0	0.0	15446.90
60	10000	-9	-9.0	0.0	18767.90
70	10000	-9	-9.0	0.0	21866.50
80	10000	-9	-9.0	0.0	25048.19
90	10000	-9	-9.0	0.0	28617.80
100	10000	-9	-9.0	0.0	31294.80

Tabela 5 – Resultados das execuções do AE.

NumF	N	E_{enc}	Me_{AE}	Var_e	T_{AE}
100	100	-7	-6.5	0.25	293.20
500	500	-9	-7.8	0.36	7595.39
10	10000	-8	7.2	0.16	2935.39
20	10000	-8	-7.5	0.25	5817.89
30	10000	-9	-7.3	0.61	9355.00
40	10000	-8	-7.6	0.44	12325.59
50	10000	-8	-7.4	0.24	15662.29
60	10000	-8	-7.4	0.64	18396.69
70	10000	-9	-7.5	0.45	21610.09
80	10000	-8	-7.6	0.24	24627.30
90	10000	-9	-7.4	0.44	27999.50
100	10000	-8	-7.4	0.24	30599.59

Como pode ser observado, o ACO obteve melhores resultados quanto a energia ótima de conformação, pois a resposta ótima foi encontrada em 100% das execuções e as médias das energias (Me_{ACO}) encontradas foram melhores neste método que no AE. O AE encontrou a energia ótima em 33,33% das execuções e suas médias não foram tão boas como o ACO, com a variância da média (Var_e) também maior que o ACO. No entanto o AE apresenta resultados um pouco melhores em relação ao tempo de execução.

A Figura 21 é uma das estruturas da proteína de 24 aminoácidos apresentada no artigo de [Huang e He \(2010\)](#). Esta figura foi escolhida pois tem a estrutura semelhante em sua forma às estruturas que mais foram encontradas nos algoritmos ACO e AE. Visualmente não se pode notar diferença nas melhores estruturas do ACO e do AE.

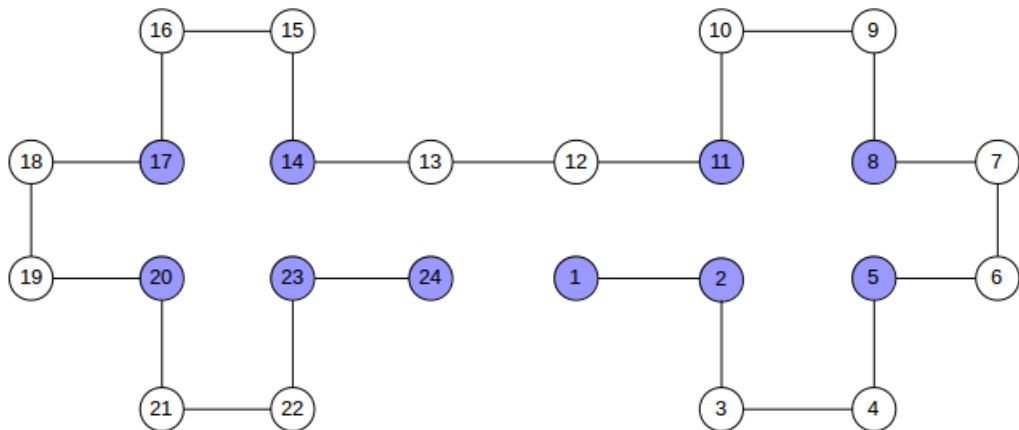


Figura 21 – Estrutura ótima da proteína de 24 aminoácidos.

Fonte: ([HUANG; HE, 2010](#))

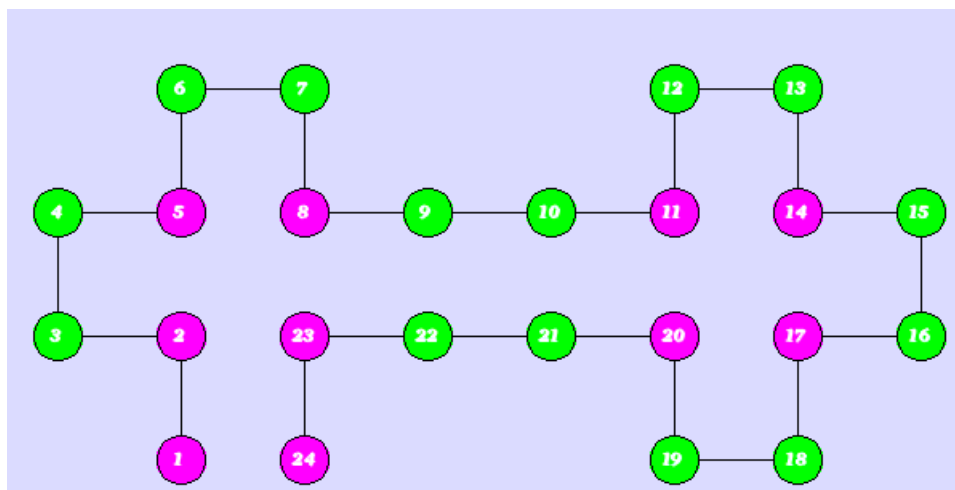


Figura 22 – Estrutura ótima obtida na execução do ACO.

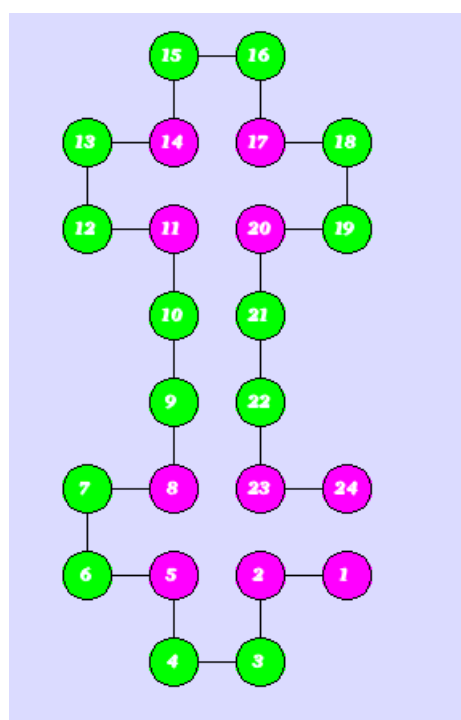


Figura 23 – Estrutura ótima obtida na execução do AE.

6.3 Resultados com a proteína de 25 aminoácidos

Nesta seção são apresentados os resultados para a proteína PPHPPHHPPPPHHP PPPHHPPPPHH. O valor ótimo para a energia de conformação desta proteína é -8.

Tabela 6 – Resultados das execuções do ACO.

NumF	N	E_{enc}	Me_{ACO}	Var_e	T_{ACO}
100	100	-7	-6.0	0.40	244.39
500	500	-7	-7.0	0.0	6185.29
15	10000	-7	-6.6	0.24	3722.30
30	10000	-7	-7.0	0.0	8204.90
40	10000	-7	-7.0	0.0	9597.70
50	10000	-7	-7.0	0.0	16612.40
60	10000	-7	-7.0	0.0	20031.80
70	10000	-7	-7.0	0.0	23162.50
80	10000	-7	-7.0	0.0	26461.40
90	10000	-7	-7.0	0.0	29883.59
100	10000	-7	-7.0	0.0	33099.89

Tabela 7 – Resultados das execuções do AE.

NumF	N	E_{enc}	Me_{AE}	Var_e	T_{AE}
100	100	-7	-5.7	0.81	172.60
500	500	-7	-6.7	0.21	4223.79
15	10000	-7	-6.3	0.61	3726.50
30	10000	-7	-6.0	0.8	5188.79
40	10000	-7	-6.4	0.64	6666.70
50	10000	-7	-6.4	0.44	14782.20
60	10000	-7	-6.5	0.45	18611.19
70	10000	-7	-6.6	0.44	21950.30
80	10000	-7	-6.7	0.41	20941.80
90	10000	-7	-5.8	0.76	27473.69
100	10000	-7	-6.2	0.76	31118.30

De acordo com o apresentado nas Tabelas 6 e 7, para esta proteína, nenhum dos dois métodos encontrou a resposta ótima. O ACO se manteve um pouco melhor que AE, pois em 81,8% das configurações atinge a energia -7 em 100% das 10 execuções, observadas pelo valor da variância. Embora o AE também tenha encontrado a mesma energia para o problema que o ACO, as médias da energia foram menores que as médias do ACO. O AE é pouco melhor que o ACO em relação ao tempo de execução, medido em milissegundos.

Para esta proteína e para as demais utilizadas neste projeto, não havia as estruturas disponíveis no artigo de [Huang e He \(2010\)](#), usado como referência.

6.4 Resultados com a proteína de 36 aminoácidos

A sequência testada foi PPPHPPHHPPPPPHHHHHHHPPHHPPPPHHPPHPP e sua energia ótima de conformação é -14.

Tabela 8 – Resultados das execuções do ACO.

NumF	N	E_{enc}	Me_{ACO}	Var_e	T_{ACO}
100	100	-13	-11.5	0.45	656.40
500	500	-13	-12.7	0.21	14343.59
10	10000	-13	-12.7	0.21	5627.50
20	10000	-13	-12.9	0.09	11268.40
30	10000	-14	-13.0	0.20	16825.19
40	10000	-14	-13.2	0.36	22607.50
50	10000	-14	-13.1	0.29	22022.30
60	10000	-14	-13.4	0.24	26208.00
70	10000	-14	-13.4	0.24	31456.00
80	10000	-14	-13.6	0.24	36649.89
90	10000	-14	-13.4	0.24	40577.89
100	10000	-14	-13.3	0.21	47396.00

Tabela 9 – Resultados das execuções do AE.

NumF	N	E_{enc}	Me_{AE}	Var_e	T_{AE}
100	100	-11	-9.8	0.76	337.20
500	500	-13	-11.5	0.45	8250.29
10	10000	-13	-11.7	1.21	3571.30
20	10000	-13	-11.6	1.24	6617.50
30	10000	-12	-10.6	0.64	10197.59
40	10000	-13	-11.0	1.4	13539.90
50	10000	-12	-10.7	1.21	16831.40
60	10000	-13	-11.2	1.16	20013.59
70	10000	-14	-11.0	1.6	22362.40
80	10000	-14	-11.4	1.64	26951.69
90	10000	-13	-11.5	1.05	30775.50
100	10000	-14	-11.1	1.89	34139.60

Como pode ser observado nas Tabelas 8 e 9, em relação a energia ótima encontrada, o ACO apresentou melhores resultados, pois foram encontradas respostas ótimas em 66,6% das execuções do algoritmo e a média de energias (Me_{ACO}) foi mais alta que o AE e o valor da variância (Var_e) é menor que o outro método. O AE atingiu a energia ótima em somente 25% das execuções. Apesar disto, o tempo de execução, em milisegundos, do AE foi bem menor em relação ao ACO, principalmente nos experimentos onde o número do indivíduos e formigas é maior.

6.5 Resultados com a proteína de 48 aminoácidos

A sequência de aminoácidos testada foi PPHPPHPPHPPPPHHHHHHHHHHH PPPPPHPPHPPHPPHPPHHHHH. A energia ótima desta proteína é -23.

Tabela 10 – Resultados das execuções do ACO.

NumF	N	E_{enc}	Me_{ACO}	Var_e	T_{ACO}
100	100	-18	-16.9	0.29	934.00
500	500	-20	-19.4	0.24	23170.90
10	10000	-20	-18.4	0.44	9059.29
20	10000	-20	-19.0	0.20	18170.19
30	10000	-20	-19.4	0.44	27272.69
40	10000	-20	-19.4	0.24	36271.89
50	10000	-20	-19.29	0.21	45585.60
60	10000	-20	-19.4	0.24	54732.00
70	10000	-20	-19.6	0.24	60194.50
80	10000	-20	-19.6	0.24	67446.00

Tabela 11 – Resultados das execuções do AE.

NumF	N	E_{enc}	Me_{AE}	Var_e	T_{AE}
100	100	-14	-12.7	1.01	251.19
500	500	-16	-15.1	0.69	6215.39
10	10000	-18	-15.3	2.61	2815.00
20	10000	-17	-14.7	1.41	5064.89
30	10000	-16	-15.2	0.36	7694.60
40	10000	-18	-16.2	1.36	12487.79
50	10000	-17	-16.4	2.84	24561.00
60	10000	-17	-15.9	3.88	28857.09
70	10000	-19	-15.6	2.04	34201.39
80	10000	-16	-15.1	0.89	38486.00

Para esta proteína foram feitas algumas modificações nos algoritmos, visto que ela é maior que as demais. No AE, o método para selecionar os pais foi substituído pelo torneio de três. Porém, os resultados na Tabela 11 são de execuções onde o AE enfatiza o *crossover* e a mutação feita na população é baixa. Na modificação do ACO, o valor da energia ótima já conhecido é fornecido ao algoritmo e o mesmo usa este valor para calcular o Δ no cálculo da atualização do feromônio, baseado em [Wilton \(2003\)](#).

Nesta proteína não se obteve a energia mínima em nenhum dos métodos. No entanto, ainda se pode notar que o ACO alcançou energias menores que o AE, com variância também menor. Em contrapartida, o AE teve média de tempo significativamente menor que o ACO.

A Tabela 12 mostra os resultados da execução do AE onde 30% dos indivíduos da população sofrem mutação. Os pais continuam sendo selecionados pelo torneio de três.

Tabela 12 – Resultados das execuções do AE.

NumF	N	E_{enc}	Me_{AE}	Var_e	T_{AE}
100	100	-17	-15.3	1.41	275.80
500	500	-19	-17.6	1.84	6954.79
10	10000	-19	-17.3	2.01	3123.39
20	10000	-19	-17.6	1.84	5696.29
30	10000	-19	-17.8	0.96	8825.90
40	10000	-20	-17.7	1.76	11196.79
50	10000	-21	-18.2	3.16	14130.40
60	10000	-21	-18.3	1.81	16874.19
70	10000	-20	-18.4	1.04	19644.09
80	10000	-19	-17.6	1.44	22370.90

Pode-se perceber que para esta nova configuração do AE, são obtidas energias melhores que no ACO, embora a ótima (-23) não tenha sido encontrada. No entanto, as médias das energias continuam mais baixas.

7 Conclusão

Analisando os dados das tabelas do Capítulo 6, conclui-se que o algoritmo ACO é bastante adequado para o problema PSP quando a busca pela energia ótima ou aproximada é o principal objetivo, mostrando resultados melhores que o AE em termos de energia. No entanto, o AE se mostra mais eficiente em relação ao tempo de execução, menor que no ACO, principalmente para proteínas maiores e em grandes quantidades de gerações e indivíduos.

Pode-se perceber pelo tempo de execução que o ACO apresenta maior custo computacional que o AE. A seguir, estão as equações dos tempos de execução do ACO e AE, respectivamente.

O tempo de execução do ACO é apresentado na Fórmula 7.1:

$$T_{ACO} = M_{fer} + M_{pop} + I * F(Const_{sol} + At_{fer} + Ord n) \quad (7.1)$$

Onde:

- M_{fer} representa a inicialização da matriz de feromônio, que tem complexidade n^2 .
- M_{pop} representa a criação da população de formigas, possuindo de complexidade $m.n$.
- $Const_{sol}$ representa a construção da solução, onde a complexidade é n^3 , incluindo os cálculos probabilísticos, o tratamento de colisões e o cálculo da energia.
- At_{fer} representa a atualização do feromônio, que tem complexidade n^2 .
- $Ord n$ representa a ordenação da população, com complexidade $n.log n$.

Assim:

$$T_{ACO} = n^2 + m.n + i.m(n^3 + n^2 + n.log n) = n^2 + m.n + i.m.n^3 + i.m.n^2 + i.m.n.log n = i.m.n^3 \quad (7.2)$$

- i é o número de iterações/gerações.
- n é o número de aminoácidos.
- m é a quantidade de formigas/indivíduos.

Pode-se concluir que $O(T_{ACO}) = O(i.m.n^3)$ pois o termo de maior expoente é o mais significativo.

O tempo de execução do AE é representado a seguir:

$$T_{AE} = M_{pop} + i.m(Repr + Calc_{energ} + Inser + Ord n) \quad (7.3)$$

Onde:

- M_{pop} representa a criação da população de indivíduos, com complexidade $n.m$.
- $Repr$ representa a criação do novo indivíduo, com complexidade n^2 , incluindo o tratamento de colisões.
- $Calc_{energ}$ indica o cálculo da energia, de complexidade n^2 .
- $Inser$ indica a inserção do filho na população, que tem complexidade $(n.m)/2$.

Assim:

$$T_{AE} = n.m + i.m(n^2 + n^2 + (n.m)/2 + n.log n) = n.m + i.m(2n^2) + i.m((n.m)/2) = 2i.m.n^2 \quad (7.4)$$

Em termos de complexidade, o termo mais significativo é aquele que apresenta o maior grau, então conclui-se que $T_{AE} = 2i.m.n^2$.

Observando as Equações 7.2 e 7.4, pode-se perceber que a complexidade de tempo de execução é maior no ACO, pois o expoente cúbico indica maior complexidade que o quadrático do AE. Para n com valores menores o ACO se comporta de modo semelhante ao AE, com um tempo de execução pouco maior. No entanto, quando o valor de n aumenta, o tempo de execução do ACO se torna praticamente inviável, demandando mais recursos computacionais, tais como memória e processador.

Para trabalhos futuros sugere-se um estudo dos parâmetros α , β e ρ do ACO a fim de descobrir novos valores que se adequam ao problema. Para o AE também é sugerido a mudança da taxa de mutação ou dos tipos de *crossover* além de estudar um método híbrido ao AE, bem como os artigos de [Lin e Su \(2011\)](#) e [Citrolo e Mauri \(2013\)](#). A computação distribuída também pode apresentar contribuições importantes tanto para a execução mais complexa do ACO quanto do AE.

Referências

- BONETTI, D. R. F. *Algoritmos de estimação de distribuição para predição ab initio de estruturas de proteínas*. Tese (Doutorado) — Universidade de São Paulo, São Carlos, mai 2015. Citado na página 24.
- BRASIL, C. R. S. *Algoritmo evolutivo de muitos objetivos para predição ab initio de estrutura de proteínas*. Tese (Doutorado) — Universidade de São Paulo, São Carlos, jul. 2012. Citado 2 vezes nas páginas 20 e 24.
- CARVALHO, E. M.; RAMOS, G. S. Otimização por colônia de formigas (ant colony optimization – aco). v. 1, 2007. Citado na página 11.
- CHU, D.; TILL, M.; ZOMAYA, A. Y. Parallel ant colony optimization for 3d protein structure prediction using the hp lattice model. In: *IPDPS*. IEEE Computer Society, 2005. ISBN 0-7695-2312-9. Disponível em: <http://dblp.uni-trier.de/db/conf/ipps/ipdps2005.html#ChuTZ05>. 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. In: GRAUDENZI, A. et al. (Ed.). *Wivace*. [s.n.], 2013. (EPTCS, v. 130), p. 61–69. Disponível em: <http://dblp.uni-trier.de/db/series/eptcs/eptcs130.html#CitroloM13>. Citado 2 vezes nas páginas 26 e 50.
- CUSTODIO, F. L.; BARBOSA, H. J. C.; DARDENNE, L. E. Investigation of the three-dimensional lattice HP protein folding model using a genetic algorithm. *Genetics and Molecular Biology*, scielo, v. 27, p. 611 – 615, 00 2004. ISSN 1415-4757. Disponível em: http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1415-47572004000400023&nrm=iso. Citado na página 24.
- CUTELLO, V.; NARZISI, G.; NICOSIA, G. A multi-objective evolutionary approach to the protein structure prediction problem. *Journal of The Royal Society Interface*, The Royal Society, v. 3, n. 6, p. 139–151, fev. 2006. ISSN 1742-5662. Disponível em: <http://dx.doi.org/10.1098/rsif.2005.0083>. Citado na página 24.
- DIAS, J. M.; BRASIL, C. R. S. Estudo das abordagens do algoritmo de colônia de formigas para o problema do caixeiro viajante. In: *Encontro Anual de Computação*. Catalão: XII EnAComp, 2015, 2015. Citado na página 27.
- DIAS, J. M.; BRASIL, C. R. S. Estudo dos parâmetros do algoritmo evolutivo aplicado ao problema de predição de estruturas de proteínas usando modelo hp-2d. In: *III FACOM TechWeek*. Uberlândia: III FACOM TechWeek, 2016. Citado na página 27.
- DORIGO, M.; BLUM, C. Ant colony optimization theory: A survey. *Theoretical Computer Science*, v. 344, n. 2–3, p. 243 – 278, 2005. ISSN 0304-3975. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0304397505003798>. Citado na página 14.
- DORIGO, M.; GAMBARDELLA, L. M. Ant colony system: A cooperative learning approach to the traveling salesman problem. *Trans. Evol. Comp*, IEEE Press,

- Piscataway, NJ, USA, v. 1, n. 1, p. 53–66, abr. 1997. ISSN 1089-778X. Disponível em: <<http://dx.doi.org/10.1109/4235.585892>>. Citado 3 vezes nas páginas 11, 14 e 15.
- GABRIEL, P. H. R. *Algoritmos evolutivos e modelos simplificados de proteínas para predição de estruturas terciárias*. Dissertação (Mestrado) — Universidade de São Paulo, São Carlos, fev. 2010. Citado na página 25.
- GOLDBERG, D. E. *Genetic Algorithms in Search, Optimization and Machine Learning*. New York, NY, USA: Addison-Wesley Longman Publishing Co., Inc., 1989. Citado 2 vezes nas páginas 10 e 12.
- HART, W. E.; ISTRAIL, S. Lattice and off-lattice side chain models of protein folding: Linear time structure prediction better than 86 *Journal of Computational Biology*, v. 4, n. 3, p. 241–259, 1997. Citado na página 21.
- HAUPT, R. L.; HAUPT, S. E. *Practical genetic algorithms*. 2. ed. New Jersey, USA: John Wiley and Sons, Inc., 2004. Citado na página 10.
- HOLLAND, J. H. *Adaptation in Natural and Artificial Systems*. [S.l.]: The University of Michigan Press, 1975. Citado 2 vezes nas páginas 10 e 12.
- HUANG, Y.; HE. Protein folding simulations of 2d hp model by the genetic algorithm based on optimal secondary structures. *Computational Biology and Chemistry*, The Royal Society, v. 34, n. 3, p. 137–142, jun. 2010. ISSN 1476-9271. Disponível em: <www.elsevier.com/locate/compbiochem>. Citado 5 vezes nas páginas 25, 38, 41, 43 e 45.
- JOHNSON, C. M.; KATIKIREDDY, A. A genetic algorithm with backtracking for protein structure prediction. In: CATTOLICO, M. (Ed.). *Gecco*. ACM, 2006. p. 299–300. ISBN 1-59593-186-4. Disponível em: <<http://dblp.uni-trier.de/db/conf/gecco/gecco2006.html#JohnsonK06>>. Citado na página 24.
- KHIMASIA, M.; COVENEY, P. Protein structure prediction as a hard optimization problem: The genetic algorithm approach. *Molecular Simulation*, v. 19, p. 205–226, 1997. Citado na página 24.
- LAU, K. F.; DILL, K. A. A lattice statistical mechanics model of the conformational and sequence spaces of proteins. *Macromolecules*, v. 22, n. 10, p. 3986–3997, 1989. Disponível em: <<http://dx.doi.org/10.1021/ma00200a030>>. Citado na página 21.
- LIANG, F.; WONG, W. H. Evolutionary monte carlo for protein folding simulations. *The Journal of Chemical Physics*, v. 115, n. 7, p. 3374–3380, 2001. Disponível em: <<http://scitation.aip.org/content/aip/journal/jcp/115/7/10.1063/1.1387478>>. Citado na página 25.
- 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, p. 140–147, jun 2011. Citado 2 vezes nas páginas 25 e 50.
- SHMYGELSKA, A.; HERNÁNDEZ, R. A.; HOOS, H. H. An ant colony optimization algorithm for the 2D HP protein folding problem. In: *Proceedings of the Third International Workshop on Ant Algorithms*. [S.l.]: Springer Verlag, 2002. p. 40–53. Citado na página 25.

SHMYGELSKA, A.; HOOS, H. An Improved Ant Colony Optimisation Algorithm for the 2D HP Protein Folding Problem. In: . [s.n.], 2003. p. 993. Disponível em: <http://dx.doi.org/10.1007/3-540-44886-1_30>. Citado na página 25.

SHMYGELSKA, A.; HOOS, H. H. An ant colony optimisation algorithm for the 2d and 3d hydrophobic polar protein folding problem. *BMC Bioinformatics*, v. 6, p. 30, 2005. Disponível em: <<http://dblp.uni-trier.de/db/journals/bmcbi/bmcbi6.html#ShmygelskaH05>>. Citado 2 vezes nas páginas 25 e 38.

STÜTZLE, T.; HOOS, H. Improvements on the ant-system: Introducing the max-min ant system. In: _____. *Artificial Neural Nets and Genetic Algorithms: Proceedings of the International Conference in Norwich*. Vienna: Springer Vienna, 1998. p. 245–249. ISBN 978-3-7091-6492-1. Disponível em: <http://dx.doi.org/10.1007/978-3-7091-6492-1_54>. Citado na página 17.

STÜTZLE, T.; HOOS, H. Max - min ant system. *Future Generation Computer Systems*, v. 16, n. 8, p. 889–914, 2000. Citado na página 38.

UNGER, R.; MOULT, J. Genetic algorithms for protein folding simulations. *Journal of Molecular Biology*, v. 231, n. 1, p. 75–81, 1993. ISSN 0022-2836. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0022283683712581>>. Citado na página 25.

WHITLEY, D. A genetic algorithm tutorial. *Statistics and Computing*, v. 4, n. 2, p. 65–85, 1994. ISSN 1573-1375. Disponível em: <<http://dx.doi.org/10.1007/BF00175354>>. Citado na página 24.

WILTON, C. *An Ant Colony Optimization Algorithm for the 2DHP Protein Folding Problem*. Dissertação (Mestrado) — University of Exeter, Exeter, sep 2003. Citado na página 47.