Comparando algoritmos de otimização computacional aplicados ao problema de predição de estruturas proteicas com modelo HP-2D

Christiane Regina Soares Brasil[1]
Júlia Manfrin Dias[2]

Resumo: Os métodos de otimização computacional são largamente aplicados a diversos tipos de problemas complexos a fim de encontrar soluções para os mesmos. Neste trabalho, os métodos de otimização estudados foram o Algoritmo Evolutivo (AE) e a Otimização por Colônia de Formiga (ACO – Ant Colony Optimization). Ambos são bioinspirados, isto é, são baseados em processos que ocorrem na natureza. Neste caso específico, o AE e o ACO foram utilizados para encontrar soluções ao desafiador problema de predição de proteínas (PSP – Protein Structure Problem), caracterizado como um problema não polinomial. Foi realizada uma comparação entre estes dois métodos aplicados ao PSP usando modelo HP-2D com algumas sequências específicas, tanto do ponto de vista computacional quanto bioquímico. Os resultados mostraram que o ACO é melhor em termo de energia, enquanto que o AE é mais adequado em termo de tempo, especialmente para proteínas maiores.

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

Abstract: ***Computational optimization methods are widely applied to several types of complex problems in order to find solutions for them. In this work, the optimization methods studied were the Evolutionary Algorithm (AE) and Ant Colony Optimization (ACO). Both are bio-inspired, that is, they are based on processes occurring in nature. In this specific case, AE and ACO were used to find solutions to the challenging protein prediction problem (PSP), also characterized as a non-polynomial problem. A comparison was made between these two methods applied to the PSP using HP-2D model with some specific sequences, from both computational and biochemical points of view. The results showed that ACO is better in terms of energy, whereas AE is more suitable in term of time, especially for larger proteins.***

Keywords: *Ant Colony Optimization, Evolutionary Algorithm, Protein Structure Problem.*

1 Introdução

No mundo real, existem inúmeros problemas complexos cuja solução não pode ser encontrada em um intervalo de tempo viável. Estes são classificados como problemas NP, ou seja, não polinomiais. Eles se caracterizam como problemas combinatórios, em que exigem uma larga busca da melhor solução em um espaço muito vasto, onde há todas as combinações das possíveis soluções [4].

Dentre estes, um dos problemas complexos e não polinomiais mais instigantes e importantes da nossa realidade é o problema de predição de estruturas de proteínas (do inglês, Protein Structure Problem – PSP) [7, 16]. As proteínas são moléculas biológicas fundamentais para a manutenção da vida em nosso planeta, pois são responsáveis por diversas funções essenciais em qualquer organismo vivo [5]. Como consequência dessa importância, percebe-se cada vez mais a necessidade do conhecimento e entendimento de suas conformações estáveis (estruturas tridimensionais), pois essas estruturas estão diretamente relacionadas às suas características

1 Faculdade de Computação, Universidade Federal do Uberlândia (UFU), MG – Brasil
{christiane.ufu@gmail.com}
2 Faculdade de Computação, Universidade Federal do Uberlândia (UFU), MG – Brasil
{juliamanfrindias@gmail.com}

http://dx.doi.org/10.5335/rbca.v9i3.7005

funcionais. Deste modo, compreendendo as funções das proteínas, pode-se gerar novas drogas ou remédios para prevenção de doenças, até hoje consideradas incuráveis, como por exemplo, o Alzheimer.

Todavia, os métodos convencionais (cristalografia e ressonância nuclear magnética) utilizados para a obtenção de estruturas de proteínas ainda não são eficientes em termo de tempo e custo. Portanto, o emprego de algoritmos de otimização para solucionar o problema PSP apresenta-se como uma boa alternativa ante aos métodos tradicionais. Podem-se citar diversas técnicas de otimização computacional aplicadas ao PSP, como Algoritmos Evolutivos [1, 2, 6, 15, 20, 23, 24], Otimização por Colônia de Formiga [10, 14, 18, 21, 25], Otimização por Enxame de Abelhas [3, 19], entre outros [26, 27]. Vale ressaltar que, em geral, usando esses métodos para problemas extremamente complexos não é obtida uma única solução (mas um conjunto de soluções); inclusive a solução ótima pode nunca ser alcançada, e nestes casos, pode ser difícil definir qual a melhor solução no conjunto das possíveis soluções.

Os algoritmos de otimização tratados neste trabalho são: o Algoritmo Evolutivo (AE) e a Otimização por Colônia de Formiga (do inglês, Ant Colony Optimization), onde os resultados destes dois métodos foram comparados entre si. As proteínas usadas neste trabalho foram extraídas dos artigos de Shmygelska [21] e Huang [15], por serem trabalhos de alta relevância.

O objetivo deste trabalho foi realizar um estudo comparativo entre estes dois métodos de otimização, em termos de energia mínima e tempos computacionais, a fim de analisar com quais parâmetros cada método seria mais eficiente para o problema de predição de estrutura de proteínas usando o modelo HP-2D.

2 Algoritmos de Otimização Computacional

Os algoritmos de otimização computacional surgiram com o propósito de encontrar uma solução ótima, ou uma solução apropriada, para um problema considerado complexo. Neste trabalho foram estudados dois destes algoritmos, o Algoritmo Evolutivo e a Otimização por Colônia de Formiga, que serão descritos a seguir.

2.1 Algoritmo Evolutivo

O Algoritmo Evolutivo (AE) é uma técnica de otimização computacional inspirada na teoria da Seleção Natural de Darwin [8]. Este algoritmo foi criado por Holland [13] e popularizado por Goldberg [11], entre os anos 1960 e 1970.

O AE mimetiza o processo biológico da evolução. Na biologia, um indivíduo representa um ser vivo e a população é um conjunto de indivíduos da mesma espécie que vivem no mesmo lugar. Deste modo, o processo de evolução consiste basicamente na manutenção dos indivíduos mais aptos para próxima geração, após o nascimento de novos indivíduos (filhos). Esses filhos possuem as características herdadas de seus pais ou de gerações anteriores, ou ainda alguma mudança genética gerada de modo aleatório. Isto ocorre devido à reprodução entre indivíduos e/ou às mutações gênicas que eles podem sofrer [11, 12].

Analogamente, no contexto computacional, há uma população de indivíduos no AE, representada por um conjunto de vetores, os quais são denominados cromossomos e que representam uma possível solução para o problema. A evolução desta população acontece como na natureza: por meio de cruzamento entre os indivíduos e/ou mutações em alguns deles.

O processo de cruzamento, ou crossover, ocorre quando há uma troca de genes entre dois indivíduos (cromossomos) e a criação de um novo indivíduo. Em geral, ocorre com dois pais, mas pode haver adaptações para mais de dois pais. Os genes trocados entre pais geram o descendente, que também pode ser um ou mais. A mutação é uma alteração que ocorre em um gene, ou mais do indivíduo. Existem vários tipos de cruzamento e mutação [11, 12], e para cada problema diferente, são aplicados de acordo com a eficiência apresentada.

A evolução do AE ocorre pela preservação dos indivíduos mais aptos (ou seja, mais bem avaliados), e o descarte dos piores indivíduos, até que se alcance o critério de parada (na maioria das vezes, o número de gerações). Os indivíduos são avaliados por uma função objetivo (fitness), que é específica para cada problema.

2.2 Otimização por Colônia de Formigas

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 feita por uma substância química, chamada feromônio. As formigas depositam essa substância no chão por onde passam, formando as trilhas de feromônio [9]. Uma vez que esses 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.

Inspirados na habilidade das formigas encontrarem caminhos ótimos, surgiram estudos com o intuito de desenvolver um método computacional similar ao comportamento das mesmas. Em 1997, foi proposta a Otimização por Colônia de Formiga (ACO) por Marco Dorigo. Deste modo, o ACO utiliza recursos que mimetizam as formigas e o feromônio. A formiga gera uma possível solução efetuando um cálculo de probabilidade, que determina o caminho que deve ser seguido. 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 variações do ACO criado por Dorigo. As mais comuns são o ACS (Ant Colony System), AS (Ant System) e MMAS (Max-Min Ant System) [9, 22]. Estas abordagens se diferenciam, na maioria das vezes, na atualização do feromônio. Neste trabalho foi utilizada a abordagem MMAS.

O cálculo de probabilidade que a formiga executa para decidir qual o caminho a ser seguido é dado pela Equação (1):

$$P_{ij} = \left(\tau_{ij}\right)^{\alpha} \left(\eta_{ij}\right)^{\beta} / \sum_{e \in N} \left(\tau_{ie}\right)^{\alpha} \left(\eta_{ie}\right)^{\beta} \quad (1)$$

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.
- Os símbolos 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.

A atualização do feromônio também ocorre por meio da Equação (2):

$$\tau_{ij} = \rho\left(\tau_{ij}\right) + \rho\Delta^{melhor} \quad (2)$$

Onde:

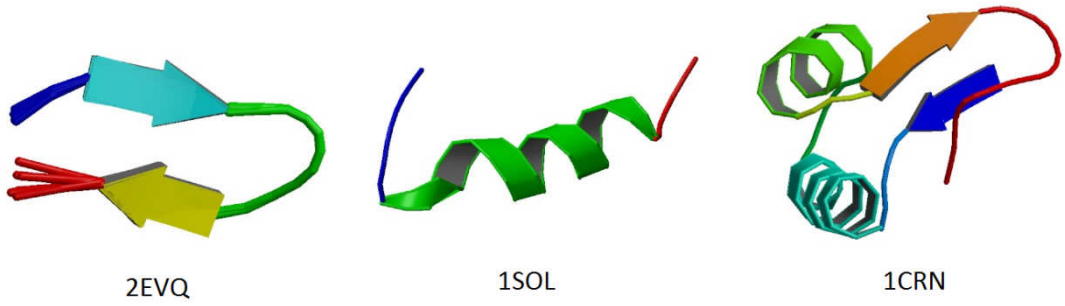
- τ_{ij} é o valor do feromônio no local pretendido.
- O símbolo ρ é o parâmetro que representa a taxa de evaporação do feromônio naquele local. Seu valor varia no intervalo de 0 a 1.
- Δ^{melhor} = E sol / E opt . E sol é a energia obtida a partir do caminho que a formiga percorreu. E opt é a melhor energia conhecida da proteína. Se esta informação for desconhecida, usa-se a melhor energia encontrada até o momento da execução. O delta (Δ) pode sofrer algumas modificações dependendo da abordagem do ACO.

3 Problema de Predição de Estruturas de Proteínas

O conhecimento da estrutura tridimensional de uma proteína (figura 1) é essencial para descoberta de suas funções, uma vez que a estrutura proteica tem uma relação direta com as suas funcionalidades [5]. Diante dessa ciência, torna-se possível a manipulação das proteínas para fins biológicos, tais como: encontrar novos

fármacos, estudar/compreender algumas doenças e elaborar novas vacinas. Portanto, encontrar a estrutura de uma proteína tem uma importância inquestionável em questões de saúde mundial. No entanto, este problema combinatório é muito difícil de ser resolvido, uma vez que sua complexidade cresce exponencialmente de acordo com o tamanho da proteína, sendo, portanto, caracterizado como um problema não-polinomial [7, 16], em termos computacionais.

Figura 1: Exemplos de estruturas tridimensionais, referentes às proteínas 2EVQ, 1SOL e 1CRN (provenientes do PDB)[3].



A **estrutura tridimensional (também chamada terciária)** se caracteriza pelo formato 3D da proteína, isto é, a conformação gráfica que a proteína é encontrada na natureza. Essa estrutura é definida pelo enovelamento (dobramento) dos aminoácidos[4] da proteína correspondente, lembrando que cada proteína possui uma sequência de aminoácidos interligados [5]. Tal enovelamento pode ser direcionado por diversos critérios, dentre eles a energia intramolecular, a energia da molécula com o meio que a envolve, a hidrofobicidade, dentre outros. A saber, os aminoácidos hidrofóbicos (hidro=água; fobia=medo, aversão) da proteína não interagem com o meio (solvente) e se voltam para o centro da estrutura, enquanto que as partes hidrofílicas ou polares da cadeia se voltam para a superfície [5].

3.1 Métodos para predição de estrutura

Existem alguns métodos para deduzir/predizer 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 viável a estes métodos ainda limitados. As abordagens computacionais para predição de estrutura proteica são ab initio, threading e homologia [5].

Neste trabalho, foi utilizada a abordagem ab initio (expressão em latim que significa “do começo”, “do princípio”), que gera modelos de estruturas tridimensionais a partir da sequência de aminoácidos da proteína e de critérios físico-químicos envolvidos no processo de enovelamento, ou seja, não é utilizado nenhum conhecimento prévio. Neste sentido, essa abordagem foi escolhida justamente por buscar a conformação de uma proteína a partir das energias importantes no seu dobramento, que neste estudo em questão foi a hidrofobicidade.

Esta abordagem utiliza-se basicamente dos princípios físico-químicos dos aminoácidos, e não em estruturas já conhecidas nem proteínas homólogas, como ocorre nas abordagens de threading e homologia. Nesta abordagem, as estruturas podem ser representadas com os seguintes modelos: full atom, lattice e off lattice. Neste trabalho, foi aplicado o modelo lattice.

Os 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

3 Protein Data Bank*:* http://www.rcsb.org/pdb/home/home.do

4 Os aminoácidos representam a menor unidade na composição de uma proteína, e apresentam na sua estrutura o grupo carboxílico (-COOH) e o grupo amina (-NH2).

movimentos 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).

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.

3.2 Modelo HP para o problema PSP

O modelo Hidrofóbico-Polar (HP) é um modelo lattice [17] de representação de proteínas proposto por Lau e Dill (1989). Esses modelos oferecem uma representação simplificada para o problema PSP, por isso 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.

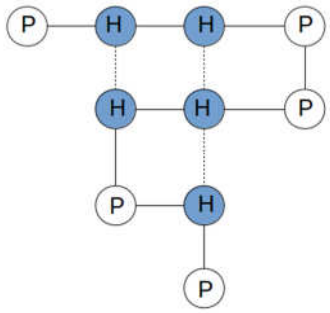
O modelo HP considera a hidrofobicidade e a polaridade dos aminoácidos para a representação computacional. A hidrofobicidade, no contexto biológico, define as moléculas que não se dissolvem nem interagem com a água, isto é, se repelem do meio aquoso, procurando se manter distantes dele. Na polaridade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

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.

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

Figura 2: Sequência proteica representada pelo Modelo HP-2D.



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 Equação (3):

$$E = \beta_{i,j} \sum \delta(r_i, r_j) \tag{3}$$

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 Implementação dos métodos AE e ACO para PSP

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

4.1 Implementação dos métodos ACO para PSP

A seguir, será descrita a implementação do ACO realizada neste trabalho.

4.1.1 Representação computacional da formiga

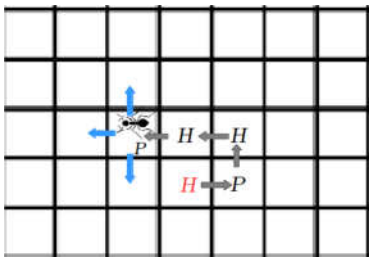
No ACO, uma população de formigas é criada inicialmente, e em seguida as formigas constroem seus caminhos pela malha 2D. Cada caminho representa uma solução para o PSP. Este caminho construído pela formiga é representado por um vetor cujas posições possuem os movimentos que a formiga realizou para construir um caminho. Nesta implementação a formiga é representada por uma estrutura que contém um vetor para movimentos, uma variável para armazenar a energia da possível solução e outro vetor para guardar as coordenadas cartesianas dos pontos do caminho. O conjunto de coordenadas são do tipo *[(x_{i}, y_{i}),...,(x_{n},y_{n})]*, onde *i = 1, 2, 3…,n*, onde *n* é o número de aminoácidos 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.

4.1.2 Movimentação da formiga

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

Suponhamos uma proteína no modelo HP com a sequência HPHHPH. De acordo com a Figura 3, 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 sítio, a formiga deve "depositar" um aminoácido e em seguida escolher qual o próximo local (indicados pelas setas azuis) para o aminoácido seguinte da cadeia.

Figura 3: Processo de formação da estrutura da proteína pelo ACO.



Nesta implementação, o ponto de partida foi o centro da malha, pois deste modo foi possível quatro movimentos para a formiga. No ponto inicial do caminho foi 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 aumentam a possibilidade de novos caminhos.

Depois do primeiro aminoácido ser colocado na malha, a formiga pode executar três movimentos, que são: ir para frente (F), virar à direita (R) ou virar à esquerda (L). Porém, ao escolher um desses movimentos, a orientação na malha é alterada. Por exemplo, na Figura 5, a formiga estava no ponto de origem sentido horizontal à direita e escolheu o movimento virar à esquerda (L). Logo após realizar esse movimento a orientação da formiga passou a ser no sentido vertical para cima. Essa orientação também precisa ser conhecida na implementação do problema.

Determinados 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 permitida a colisão, isto é, que dois aminoácidos ocupem o mesmo lugar na malha simultaneamente. Para prevenir essa colisão, a formiga analisa 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 retorna para a posição anterior. Esse processo de backtracking pode fazer a formiga voltar quantas vezes forem necessárias, até encontrar uma posição em que não dê possibilidades de ocorrer colisões.

4.1.3 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.

A princípio, 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 (Figura 5). Se o aminoácido a ser colocado neste ponto for do tipo H, ela olhará para todos os vizinhos de F_{pos}. 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.

Deste modo, a probabilidade de escolher a próxima posição é calculada pela Equação (4):

$$Pij = (\tau_{ij})^{\alpha} (\eta_{ij})^{\beta} / \sum (\tau_{ie})^{\alpha} (\eta_{ie})^{\beta} \quad (4)$$

Onde:

- 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.

4.1.4 Matriz de feromônios

Após a formiga concluir seu caminho pela malha e uma possível estrutura ser encontrada, ela atualiza seu feromônio. Neste trabalho, a matriz de feromônios corresponde à malha bidimensional, onde cada posição da matriz de feromônio é associado à posição corresponde da malha. Deste modo, as posições que a formiga percorreu na malha são as mesmas posições em que é depositado feromônio na matriz de feromônios. No começo, o valor do feromônio é 1 para todas as posições.

4.1.5 Atualização do feromônio

Ao finalizar a construção do caminho na malha, a formiga deposita uma certa quantidade de feromônio no mesmo. A quantidade de feromônio pode ficar alta quando há muitas escolhas pelo mesmo caminho. Em contrapartida, quando o percurso não representa uma boa solução as demais formigas não seguem aquele caminho e, por isso, a quantidade de feromônio presente ali sofre uma evaporação.

A implementação descrita neste trabalho 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. Neste caso, escolheu-se a melhor formiga encontrada até o momento.

O depósito de feromônio considera a evaporação e é dado pela Equação (5):

$$\tau_{ij} = (\rho)(\tau_{ij}) + \rho\,\Delta^{melhor} \quad (5)$$

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.

4.2 Implementação do método AE para PSP

A seguir, será descrita a implementação do AE realizada neste trabalho.

4.2.1 Representação computacional do indivíduo

No AE desenvolvido neste trabalho, o indivíduo foi representado por um vetor, cujas suas posições são denominadas genes. Desta maneira, a população de indivíduos foi armazenada em uma matriz.

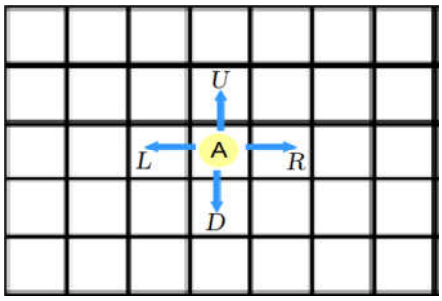
Considerando o AE para resolver o PSP, o vetor de cada indivíduo guardou uma sequência de movimentos para cada um dos aminoácidos da proteína, com exceção do primeiro aminoácido da cadeia. Os movimentos que a cadeia pode fazer na malha são: ir à direita (R), ir à esquerda (L), ir para cima (U) e ir para baixo (D). O funcionamento destes movimentos na malha será explicado mais adiante.

4.2.2 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. Como foi dito, 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.

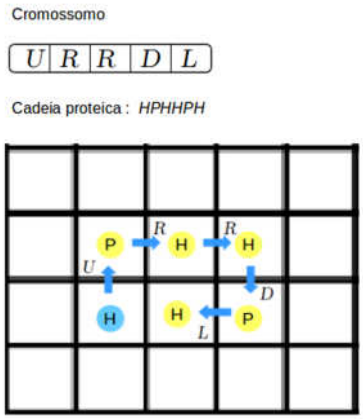
Na Figura 4, considere A como sendo o primeiro aminoácido da cadeia proteica. O segundo aminoácido será alocado em uma das quatro posições indicadas pelas setas azuis. A posição correta será definida pela informação que está no gene do indivíduo, que foi escolhida aleatoriamente em uma das quatro posições (R, L, U, D). Os demais aminoácidos seguem a mesma lógica.

Figura 4: Direções de locomoção dos aminoácidos.



O processo de construção da estrutura começa com o primeiro aminoácido da cadeia colocado em um ponto na malha (tal como no ACO, foi o centro da malha). 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 5 mostra como a sequência de aminoácidos se dobra na malha de acordo com a informação do cromossomo.

Figura 5: Formação da estrutura proteica a partir da informação do cromossomo.



Os indivíduos podem representar soluções que apresentam colisões, assim como no ACO. No entanto, diferente do ACO, a colisão não é evitada. No AE implementado, 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.

4.3 Operações de crossover, mutação e seleção dos pais

Antes de ocorrer, de fato, a reprodução, é necessário 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 [5]. 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 tipos de crossovers: os de 1-ponto e 2-pontos [5]. No crossover de 1-ponto os pais são divididos ao meio e os filhos se originam com uma parte do primeiro pai e com a outra parte do segundo pai. O crossover de 2-pontos ocorre de modo semelhante, porém os cromossomos dos pais sofrem dois “cortes”, sendo dividido em três partes. Os filhos são gerados com a parte do meio do cromossomo de um pai e com as extremidades do cromossomo vindas do outro pai.

Os dois tipos de crossover presentes 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. Essas taxas de probabilidades de crossover foram definidas empiricamente.

A mutação desenvolvida neste trabalho foi baseada no trabalho de Huang [15]. 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. A fim de preservar os melhores indivíduos ao longo das gerações, aplica-se o elitismo [5]. O elitismo consiste em manter os melhores indivíduos

de uma geração e transferi-los para a geração seguinte. Nesta implementação a quantidade de indivíduos mantidos para a próxima geração é equivalente à metade da população, ou seja, um elitismo de 50%.

5 Resultados

Nesta seção serão apresentados os resultados comparativos entre os dois métodos (ACO e AE) para resolver o problema PSP. Ambos utilizaram as proteínas advindas dos artigos de Shmygelska e Hoos [21] e Huang [15], alcançando as mesmas energias mínimas do material de referência em três casos dos quatro testados.

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 ρ 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 equação 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 [15]. Essas operações foram utilizadas no AE do seguinte modo: sendo N o número de gerações, da geração 1 até 0.1*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*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%. Essas taxas de probabilidades de crossover e mutação foram definidas empiricamente. A escolha dos pais foi feita de forma aleatória entre todos os indivíduos da população para as proteínas de 20 a 36.

As tabelas a seguir mostram os resultados de execuções do ACO e do AE, onde foram variados os números de indivíduos, formigas, gerações e iterações. Cada linha apresenta o resultado de 10 execuções, de acordo com o número de indivíduos, formigas e gerações e interações.

5.1 Resultados com a proteína de 20 aminoácidos

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

Tabela 1: 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**
100	**10000**	**-9**	**-9.0**	**0.0**	**29323.50**

Tabela 2: 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
100	10000	-9	-8.2	0.36	24475.30

Note que 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 1 e 2, 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}). 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 relativamente menor que o ACO, em todas as configurações.

5.2 Resultados com a proteína de 24 aminoácidos

A proteína de sequência HHPPHPPHPPHPPHPPHPPHPPHH foi avaliada e sua energia ótima é -9. As tabelas abaixo apresentam os resultados das execuções do ACO e do AE, respectivamente.

Tabela 3: 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
100	**10000**	**-9**	**-9.0**	**0.0**	**31294.80**

Tabela 4: Resultados das execuções do AE.

NumInd	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
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 na configuração 100-10000 e as médias das energias (Me_{ACO}) encontradas foram melhores neste método que no AE, de modo geral. O AE encontrou a energia ótima em 40% 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 melhores em relação ao tempo de execução, repetindo o evento da proteína de 20 aminoácidos.

5.3 Resultados com a proteína de 36 aminoácidos

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

Tabela 5: 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
100	**10000**	**-14**	**-13.3**	**0.21**	**47396.00**

Tabela 6: Resultados das execuções do AE.

NumInd	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
100	**10000**	**-14**	**-11.1**	**1.89**	**34139.60**

Como pode ser observado nas Tabelas 5 e 6, em relação a energia ótima encontrada, o ACO apresentou melhores valores para a média de energias (Me_{ACO}) que o AE, o que significa que os valores encontrados para a energia, em 10 execuções, foram mais próximas da ótima (-14). Apesar disto, o tempo de execução, em

milissegundos, do AE foi significativamente menor em relação ao ACO, principalmente nos experimentos onde o número dos indivíduos e formigas é maior.

6 Conclusões

Analisando os resultados obtidos neste trabalho, 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. Isto é justificado pelo fato do AE apresentar um tempo computacional da ordem quadrática em relação a N, onde N é o número de aminoácidos, enquanto que o ACO tem tempo computacional da ordem cúbica em relação a N. Para N com valores pequenos, 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. Vale ressaltar que ambos os métodos atingiram as energias mínimas encontradas nos artigos de referência em praticamente todos os casos [15, 21].

Para trabalhos futuros, sugere-se um estudo dos parâmetros α, β e ρ do ACO, a fim de descobrir novos valores para melhor adequação 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, como os artigos de Lin e Su [19] e Turabien [24]. A computação distribuída também pode apresentar contribuições importantes, principalmente para os casos mais complexos, tanto para o ACO quanto do AE.

Referências

[1] 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, 2012.

[2] BRASIL, C. R. S.; DELBEM, A. C. B.; DA SILVA, F. L. B. Multiobjective evolutionary algorithm with many tables for purely ab initio protein structure prediction. *Journal of Computational Chemistry*, v. 34, p. 1719-1734, 2013.

[3] CHENG-JIAN LIN; SHIH-CHIEH SU. Using an efficient artificial bee colony algorithm for protein structure prediction on lattice models. *International Journal of Innovative Computing, Information and Control.* v. 8, n. 3(B), p. 2049-2064, 2012.

[4] CORMEN, T. H., LEISERSON, C. E., RIVEST, R. L., STEIN, C. *Introduction to Algorithms*, Third Edition. MIT Press and McGraw-Hill. Section 34: NP-Completeness, p. 1048.

[5] COX, M.; DOUDNA, J. A. *Biologia Molecular: Princípios e Técnicas*. Artmed Editora, 2012.

[6] CUSTÓDIO, F. L. *Algoritmos genéticos para predição Ab Initio de Estruturas de Proteínas*, PhD thesis, LNCC, Petrópolis-RJ, 2008.

[7] CRESCENZI, P., GOLDMAN, D., PAPADIMITRIOU, C.H., PICCOLBONI, A., YANNAKAKIS, M. On the Complexity of Protein Folding. *Journal of Computational Biology.* v. 50, p. 423–466, 1998.

[8] DARWIN, C. R. *On the Origin of Species*, 1859.

[9] DORIGO, M.; GAMBARDELLA, L. M. Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*. Press, Piscataway, NJ, USA, v. 1, n. 1, p. 53–66, abr. 1997. ISSN 1089-778X.

[10] FIDANOVA, S.; LIRKOV, I. Ant colony system approach for protein folding. *In: Procceedings of the International Multiconference on Computer Science and Information Technology*. Pp. 887-891. 2008.

[11] GOLDBERG, D. E. *Genetic Algorithms in Search, Optimization and Machine Learning*. New York, NY, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.

[12] HAUPT, R. L.; HAUPT, S. E. *Practical genetic algorithms*. 2. ed. New Jersey, USA: John Wiley and Sons, Inc., 2004.

[13] HOLLAND, J. H. *Adaptation in Natural and Artificial Systems*. [S.l.]: The University of Michigan Press, 1975.

[14] HU, X.; ZHANG, J.; XIAO, J.; LI, Y. Protein Folding in Hydrophobic-Polar Lattice Model: A Flexible Ant-Colony Optimization Approach. *Protein and Peptide Letters*, v. 15, n. 5, p. 469 - 477, 2008.

[15] 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.

[16] KHIMASIA, M.; COVENEY, P. Protein structure prediction as a hard optimization problem: The genetic algorithm approach. *Molecular Simulation*, v. 19, p. 205–226, 1997.

[17] 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.

[18] LLANES, A., VELEZ, C., SANCHEZ, A. M., SANCHEZ, H., CECILIA, J. M. Parallel Ant Colony Optimization for the HP Protein Folding Problem. International Conference on Bioinformatics and Biomedical Engineering. IWBBIO 2016: *Bioinformatics and Biomedical Engineering*. p. 615-626, 2016.

[19] LIN, Cheng-Jian; SU, Shih-Chieh - Protein 3D HP Model Folding Simulation Using a Hybrid of Genetic Algorithm and Particle Swarm Optimization, *International Journal of Fuzzy Systems*, Vol. 13, No. 2, 2011.

[20] GABRIEL, P. H. R., MELO, V. V. , DELBEM, A. C. B. Algoritmos evolutivos e modelo HP para predição de estruturas de proteínas. *Revista Controle & Automação*/Vol.23 no.1/Janeiro e Fevereiro, 2012.

[21] SHMYGELSKA, A. and HOSS, H. H. An ant colony optimization algorithm for the 2D and 3D hydrophobic polar protein folding problem. *BMC Bioinformatics*. 6(30): 1–22, 2005.

[22] STÜTZLE, T.; HOOS, H. Max - min ant system. *Future Generation Computer Systems*, v. 16, n. 8, p. 889–914, 2000.

[23] TSAY, J. and SU, S. An effective evolutionary algorithm for protein folding on 3D FCC HP model by lattice rotation and generalized move sets. *Proteome Science*. 2013; 11(Suppl 1): S19. 2013.

[24] TURABIEN, H. A Hybrid Genetic Algorithm for 2D Protein Folding Simulations. *International Journal of Computer Applications*. v. 139, n. 3, 2016.

[25] WILTON, C. *An Ant Colony Optimization Algorithm for the 2DHP Protein Folding Problem*. Dissertação (Mestrado) — University of Exeter, Exeter, 2003.

[26] ZHANG, Y.; WU, L. WANG, S. Solving Two-Dimensional HP Model by Firefly Algorithm and Simplified Energy Function. *Mathematical Problems in Engineering*. v. 2013, 9 pages, 2013.

[27] ZHAO, X. Advances on protein folding simulations based on the lattice HP models with natural computing. *Applied Soft Computing*. v. 8, n. 2, p. 1029–1040, 2008.