

Algoritmos multiobjetivos aplicados ao problema de predição de estruturas proteínas

Vidal Daniel da Fontoura, Ricardo Remes de Lima, Aurora Trinidad Ramirez Pozo e Roberto Santana Hermida

Resumo—Diversos métodos heurísticos vêm sendo aplicados com sucesso à predição de estruturas de proteínas utilizando modelos simplificados para a representação do problema. Estes métodos geralmente buscam a estrutura de uma proteína em que a energia livre seja mínima. Este estudo avalia a aplicação de algoritmos genéticos multiobjetivo combinados com uma representação relativa para o modelo de predição 2D-HP. Experimentos foram conduzidos com dois algoritmos bem conhecidos: Non-dominated sorting Genetic Algorithm II (NSGAI) e Indicator-Based Evolutionary Algorithm (IBEA). Os algoritmos foram avaliados e comparados utilizando diferentes operadores genéticos.

I. INTRODUÇÃO

Uma proteína é uma macromolécula composta por uma sequência de aminoácidos. Sua função biológica é determinada pelo molde de sua estrutura. Entender como as proteínas se moldam é de grande importância para a biologia, bioquímica e medicina. Entretanto não é possível determinar a estrutura exata de uma proteína utilizando o modelo atômico analítico completo, mesmo com os mais poderosos recursos computacionais [1].

Diante deste cenário, diferentes abordagens heurísticas vêm sendo desenvolvidas para reduzir a complexidade computacional envolvida na determinação da estrutura de proteínas. Dentre elas, existem estudos que exploram a predição de estrutura de proteínas combinadas com algoritmos evolutivos como, por exemplo, Li et al. [2] que propõem um AE (Algoritmo Evolucionário) mono-objetivo em conjunto com uma estratégia de busca local para o problema PSP utilizando o modelo simplificado Hidrofóbico-Polar (HP). Já Soares et al. [3] propõem um algoritmo multiobjetivo em tabelas e comparam seu desempenho em relação ao NSGAI [4], otimizando duas funções de energia de grande importância no processo de conformação: van der Waals e eletrostática. Gabriel et al. [5] também propõem a utilização de um algoritmo multiobjetivo em tabelas similar ao proposto em [3], porém utilizando o modelo HP para representação e avaliação das soluções.

Este trabalho propõem a aplicação e comparação dos algoritmos genéticos multiobjetivo NSGAI e IBEA [6] com o primeiro objetivo de minimizar a energia calculada a partir do modelo HP e um segundo objetivo que busca minimizar a distância euclidiana entre aminoácidos de uma proteína.

O artigo está organizado da seguinte maneira: na seção II é apresentado o problema PSP, modelo HP, e a representação relativa para o modelo HP. A seção III discorre sobre algoritmos genéticos e otimização multiobjetivo. A seção IV apresenta os métodos comparados. A seção V explica os experimentos que foram conduzidos. A seção VI apresenta

os trabalhos correlatos. Por fim a seção VII conclui a análise sobre os experimentos e menciona os possíveis incrementos a este estudo.

II. PROBLEMA DE PREDIÇÃO DE ESTRUTURAS DE PROTEÍNAS

O problema PSP trata da determinação da estrutura final de uma proteína utilizando apenas informação referente a sua sequência de aminoácidos [7]. Uma abordagem computacional para prever a estrutura de uma proteína necessita de um modelo que a represente de maneira abstrata num certo nível de detalhamento. Baseada nas leis da termodinâmica a predição é modelada a partir da minimização da energia livre correspondente às possíveis conformações que uma proteína pode obter. Formalmente a conformação nativa de uma proteína é definida como a conformação em que a energia livre é mínima. O nível de detalhamento dos modelos pode variar. Por exemplo, a estrutura da proteína poderia ser representada apenas pelos átomos que compõem o *backbone* desconsiderando ramificações laterais, por uma representação espacial de todos os seus átomos, por todos os átomos exceto o hidrogênio ou apenas os elementos hidrofóbicos-polares representados em rede[1].

A. Modelo HP

O modelo Hidrofóbico-Polar, ou modelo HP, desenvolvido por Lau e Dill [8], é um modelo em rede, dentre outros, que é utilizado nas pesquisas para PSP, e, assim como os demais, é um modelo de representação de proteínas que possui simplificações. O modelo considera apenas dois tipos de resíduos: o Hidrofóbico (H) e o Hidrofílico ou Polar (P). Uma proteína é constituída por uma sequência destes resíduos posicionados em uma grade (do inglês *lattice*) regular, de forma a construir um caminho que não se cruza [9]. Neste modelo, o principal conceito utilizado é o conceito de vizinhança. Dados dois resíduos, eles são considerados vizinhos caso sejam adjacentes na grade e na sequência (vizinhos ligados) ou apenas adjacentes na grade (vizinhos topológicos). A Figura 1 apresenta um exemplo de uma sequência no modelo HP, onde os círculos pretos representam os resíduos H e, os brancos resíduos P. Vizinhos ligados estão representados pela linha contínua, enquanto que vizinhos topológicos estão circulos com uma elipse tracejada.

No modelo HP, dependendo do par de resíduos que formam um contato topológico (vizinhos topológicos), são atribuídos os seguintes valores: $HH=-1$ e $HP = PP = 0$. A energia livre de uma conformação é inversamente proporcional ao número

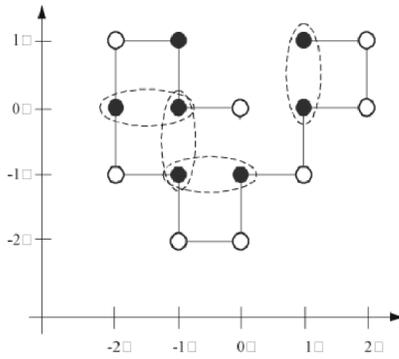


FIGURA 1: Conformação hipotética de uma cadeia de 15 aminoácidos utilizando modelo 2D-HP. Adaptado de [1]

de vizinhos topológicos HH. Consequentemente, minimizar a energia livre é equivalente a maximizar a quantidade de vizinhos topológicos onde o par é composto por resíduos hidrofóbicos (HH) [1].

B. Representação relativa

A representação relativa para o modelo HP se baseia na construção da estrutura da proteína utilizando uma sequência de direções. Nesta abordagem, cada resíduo tem sua posição definida relativamente ao resíduo anterior [1]. Para uma cadeia de N resíduos, a partir do primeiro que é fixo, todo resíduo subsequente é posicionado de acordo com uma dada direção. Para o modelo 2D-HP apenas três direções são possíveis:

- Em frente: continuar na mesma direção em que estava;
- Virar à direita: posicionar o resíduo à direita do anterior;
- Virar à esquerda: posicionar o resíduo à esquerda do anterior.

A cada resíduo que é colocado, a direção atual é atualizada. Desta forma, os comandos “seguir em frente”, “virar à direita” e “virar à esquerda” vão estar todos baseados no resíduo anterior, como mostra a figura 2.

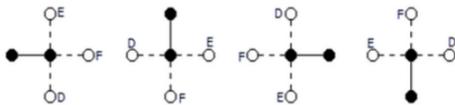


FIGURA 2: Representação relativa, onde F significa um movimento em frente, D movimento a direita e E um movimento a esquerda.

III. ALGORITMOS EVOLUCIONÁRIOS E OTIMIZAÇÃO MULTIOBJETIVA

Algoritmos Evolucionários (EAs - Evolutionary Algorithms) são uma técnica de busca e otimização, altamente paralela, inspirada no princípio Darwiniano de seleção natural e reprodução genética. Os princípios da natureza nos quais os EAs se inspiram são simples. De acordo com a teoria de

C. Darwin, o princípio de seleção privilegia os indivíduos com maior aptidão, portanto, com maior probabilidade de reprodução. Indivíduos com mais descendentes têm mais chance de perpetuarem seus códigos genéticos nas próximas gerações. Tais códigos genéticos constituem a identidade de cada indivíduo e estão representados nos cromossomos. Estes princípios são imitados na construção de algoritmos computacionais, que buscam uma melhor solução para um determinado problema através da evolução de populações de soluções codificadas por cromossomos artificiais – estrutura de dados utilizada para representar uma possível solução para o problema na execução do algoritmo [10]. Problemas do mundo real costumam ter múltiplos objetivos a serem minimizados/maximizados e estão presentes na maioria das áreas do conhecimento. Para otimizar problemas multiobjetivos são considerados 2 ou mais objetivos que costumam ser conflitantes entre si. Para estes problemas não é possível encontrar uma única solução. Um conjunto de soluções ótimas é alcançado avaliando a relação de dominância de Pareto [11] entre as soluções. O objetivo é encontrar as soluções que não são dominadas por nenhuma outra. Uma solução domina outra, se e somente se, for melhor em pelo menos um dos objetivos, sem ser pior em nenhum dos outros objetivos. O conjunto de soluções não dominadas formam a fronteira de Pareto. Encontrar a fronteira de Pareto real é NP-Difícil [12], portanto o objetivo é encontrar uma boa aproximação desta fronteira. MOEAs (Multi-Objective Evolutionary Algorithms) são extensões de AEs para problemas multiobjetivos que aplicam os conceitos de dominância de Pareto para criar estratégias distintas para evoluir e diversificar as soluções. Neste trabalho foram utilizados dois MOEAs: NSGAI [4] e IBEA [6].

A. NSGAI (Non-dominated sorting Genetic Algorithm II)

Este algoritmo tem como principal característica um forte mecanismo de elitismo, classificando, a cada geração, todas as soluções em diferentes *fronts* de acordo com a relação de não-dominância (linha 15 do Algoritmo 1). Após a classificação, soluções pertencentes ao primeiro *front*, são não dominadas por nenhuma outra solução, soluções pertencentes ao segundo *front* são dominadas apenas pelas soluções contidas no primeiro *front*, da mesma maneira soluções do terceiro *front* são dominadas por soluções do segundo e do primeiro *front* e assim por diante. Para soluções de uma mesma *front* o NSGAI utiliza o operador de *crowding distance*, para calcular o quão distante estão os vizinhos de uma determinada solução (linha 19 do Algoritmo 1). Soluções com o valor de *crowding distance* maior são priorizadas pois irão contribuir com uma maior diversidade para a população (linha 23 e 24 do Algoritmo 1). Ambas classificações são utilizadas pelo operador de seleção (linha 24 do Algoritmo 1). O torneio binário seleciona soluções do menor *front* e com maiores valores de *crowding distance*. Uma nova população é gerada utilizando os operadores de cruzamento e mutação (linha 25 do Algoritmo 1).

Algoritmo 1: NSGAI

Entrada:

N // Tamanho da População

T // Quantidade máxima de avaliações

Saída:

P // Conjunto de soluções não dominadas

```
1 inicio
2    $P_0 \leftarrow \text{Inicilizar\_Populacao}(n);$ 
3    $\text{AtribuicaoFitness}(P_0);$ 
4    $\text{FastNonDominatedSort}(P_0);$ 
5    $Q_0 \leftarrow 0;$ 
6   enquanto  $Q_0 < N$  faça
7      $\text{Parents} \leftarrow \text{TorneioBinario}(P_0);$ 
8      $\text{Children} \leftarrow \text{CruzamentoMutacao}(\text{Parents});$ 
9      $Q_0 \leftarrow \text{Children};$ 
10  fim
11   $\text{AtribuicaoFitness}(Q_0);$ 
12   $t \leftarrow 0;$ 
13  enquanto  $t < T$  faça
14     $R_t \leftarrow P_t \cup Q_t;$ 
15     $\text{Fronts} \leftarrow \text{FastNonDominatedSort}(R_t);$ 
16     $P_{t+1} \leftarrow 0;$ 
17     $i \leftarrow 0;$ 
18    enquanto  $P_{t+1} + \text{Front}_i < N$  faça
19       $\text{CrowdingDistanceAssignment}(\text{Front}_i);$ 
20       $P_{t+1} \leftarrow P_{t+1} \cup \text{Front}_i;$ 
21       $i \leftarrow i + 1;$ 
22    fim
23     $\text{CrowdingDistanceSort}(\text{Front}_i);$ 
24     $P_{t+1} \leftarrow P_{t+1} \cup \text{Front}_i[1 : (N - P_{t+1})]$ 
25    //Preenche  $P_{t+1}$  com as melhores soluções da
26    // $\text{Front}_i$  segundo o critério de crowding-distance
27     $\text{Parents} \leftarrow \text{TorneioBinario}(P_{t+1});$ 
28     $Q_{t+1} \leftarrow \text{CruzamentoMutacao}(\text{Parents});$ 
29     $t \leftarrow t + 1;$ 
30  fim
```

B. IBEA (Indicator-Based Evolutionary Algorithm)

No contexto de otimização multiobjetiva, otimizar consiste em encontrar uma fronteira ótima com uma boa aproximação da fronteira de Pareto. Contudo não existe uma definição geral do que é uma boa aproximação. Dessa forma, indicadores costumam ser utilizados para avaliar a qualidade de uma fronteira aproximada. O ¹*hypervolume* é um exemplo de indicador para avaliação e comparação de fronteiras. O IBEA é um algoritmo que considera a otimização de indicadores de qualidade. O indicador é a maneira de avaliar o conjunto não dominado de soluções [13]. Para utilizar o IBEA é necessário definir o indicador que será utilizado para associar cada par ordenado de soluções a um valor escalar. Um dos indicadores mais utilizados é o *hypervolume* devido a sua capacidade de avaliar a convergência e diversidade ao mesmo tempo no processo de busca [14].

Dada uma população P que representa uma amostra do espaço de busca e um indicador I , a aptidão (*fitness*) de uma solução é dada de acordo com a sua contribuição, em relação aos objetivos, ao processo de otimização (linha 3 do algoritmo 2). A função de aptidão F utilizada pelo IBEA é definida pela equação 1.

$$F(x_i) = \sum_{x_j \in (P - x_i)} -e^{-\frac{I_{Hy}(x_j, x_i)}{k}} \quad (1)$$

onde k é um parâmetro normalmente ajustado para 0,05. O valor $F(x_i)$ corresponde a uma medida da perda de qualidade da aproximação da fronteira de Pareto se a solução x_i for removida da população [13] baseado no valor do indicador qualidade I_{Hy} , neste caso *hypervolume*. Baseado no cálculo de *fitness* descrito acima o algoritmo básico do IBEA consiste em realizar de forma iterativa a seleção (linha 10 do algoritmo 2), cruzamento, mutação (linha 11 do algoritmo 2) e seleção ambiental, removendo o pior indivíduo da população e atualizando os valores de aptidão dos indivíduos restantes (linhas 4 a 8 do algoritmo 2).

Algoritmo 2: IBEA

Entrada:

N // Tamanho da População

T // Quantidade máxima de avaliações

k //Fator de escalonamento do fitness

Saída:

P // Conjunto de soluções não dominadas

```
1 Criação de uma população inicial  $P$  de tamanho  $N$ 
2 Inicializa o contador  $m \leftarrow 0$ 
3 Atribui valor fitness do indivíduos contidos em  $P$ 
  utilizando a equação descrita na 1
4 enquanto  $P > N$  faça
5   Seleciona o indivíduo  $x^*$  com o menor valor de
   fitness
6   Remover  $x^*$  da população  $P$ 
7   Atualizar os valores de fitness dos indivíduos
   restantes utilizando a equação apresentada na figura
   3, sendo  $x_j$  igual a  $x^*$ , para todo  $x \in P$ .
8 fim
9 Se  $m \geq T$  ou outro critério de parada for alcançado
   $A \leftarrow P$ . Execução termina e  $A$  é retornado.
10 Seleciona indivíduos via torneio binário com troca da
   população  $P$  para gerar uma população temporária  $\bar{P}$ 
11 Os operadores de cruzamento e mutação são aplicados
   em  $\bar{P}$  e adicionado os descendentes resultantes em  $P$ .
   Incrementa ( $m = m + 1$ ) e a execução retorna a linha 3.
```

¹*Hypervolume* : Indicador de qualidade proposto e utilizado inicialmente no estudo [15], denotado como o "tamanho do espaço de busca coberto". Este indicador tem duas vantagens importantes em relação a outros indicadores: [16] 1 - Sensitivo a qualquer tipo de melhoria de conjunto de aproximação em relação a outro conjunto. 2 - Como resultado de 1, o indicador garante que para qualquer conjunto de aproximação A que obtenha os maiores valores de *hypervolume* contém todas as soluções da fronteira ótima de Pareto.

IV. MÉTODOS COMPARADOS

O método proposto consistiu na aplicação dos algoritmos NSGAI e IBEA ao problema PSP utilizando a representação relativa aplicada o modelo HP. O framework de otimização multiobjetiva jMetal [17] foi utilizado pois contém implementações para MOEAs utilizados. Para avaliar as soluções geradas pelos algoritmos, um mecanismo de avaliação foi implementado levando em consideração dois objetivos:

- I Maximizar a quantidade de vizinhos topológicos HH (objetivo principal)
- II Minimizar a máxima distância euclidiana entre resíduos (objetivo secundário)

O primeiro objetivo guia a busca com intenção de encontrar a solução que gere uma estrutura onde o valor de energia seja mínimo (maximizando a quantidade de vizinhos topológicos HH), dessa maneira, obtendo uma estrutura mais próxima da conformação nativa de uma proteína. Já o segundo objetivo permite diferenciar soluções com o mesmo valor de energia mas com graus de compactação diferentes, pois quanto menor for o valor máximo para distancia euclidiana, mais compacta será a conformação gerada.

Os cromossomos foram representados por vetores inteiros onde os genes especificam em qual direção, relativa ao resíduo anterior, deve ser posicionado o próximo resíduo. Os genes podem assumir um dos valores (0, 1, 2), onde 0 indica que o próximo resíduo deve ser posicionado a direita do resíduo anterior, 1 indica que o próximo resíduo deve ser posicionado a frente do resíduo anterior e 2 indica que o próximo resíduo deve ser posicionado a esquerda do resíduo anterior. A figura 3 demonstra um exemplo de um possível cromossomo para uma cadeia de 10 resíduos e a conformação que é gerada por ele.

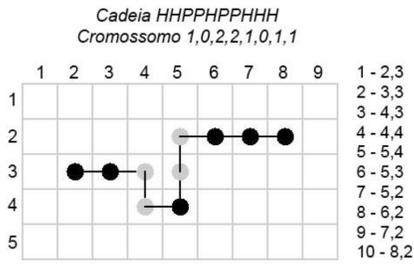


FIGURA 3: Exemplo de conformação gerada a partir de um cromossomo utilizando a representação relativa.

A representação relativa está sujeita à geração de soluções infactíveis ao problema utilizando o modelo HP. Uma solução é considerada infactível quando um resíduo 'colide' com outro resíduo que já havia sido posicionado na grade. Um mecanismo de reparação simples foi desenvolvido a fim de contornar o problema de geração de soluções infactíveis e pode ser observado no algoritmo 3.

Este mecanismo foi implementado pois foi observado em experimentos prévios que o numero de soluções infactíveis era

Algoritmo 3: Mecanismo de reparação de soluções infactíveis

- 1 Obtem a direção em que o próximo resíduo deve ser posicionado.
- 2 Verifica se esta direção causará uma colisão.
- 3 Caso a colisão seja identificada uma nova direção é utilizada.
- 4 Repetir os passos 2,3 até que seja possível posicionar o próximo resíduo ou se todas as direções foram testadas e nenhuma não causa colisão.
- 5 Se foi possível posicionar o próximo resíduo o mecanismo de reparação obteve sucesso, se por acaso não foi possível a solução é considerada infactível e será penalizada no processo de avaliação.

muito grande. É necessário mencionar que mesmo utilizando o mecanismo de reparação soluções infactíveis ainda podem ser geradas pois o mecanismo nem sempre consegue reparar. Contudo soluções infactíveis são penalizadas subtraindo a quantidade de colisões da quantidade de vizinhos topológicos HH.

Para avaliar e comparar o desempenho dos algoritmos multiobjetivos, indicadores de qualidade costumam ser utilizados. Neste estudo foi utilizado o indicador *hypervolume*, o qual considera o volume do espaço de busca dominado por uma fronteira ótima[18]. Quanto maior o *hypervolume* melhor é a qualidade da fronteira ótima encontrada por um algoritmo.

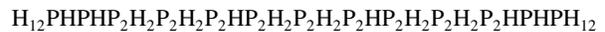
V. EXPERIMENTOS E RESULTADOS

Experimentos foram conduzidos utilizando os algoritmos NSGAI e IBEA apresentados na seção IV. Uma variação e combinação de parâmetros foi realizada para encontrar a melhor configuração de cada algoritmo. Resultando em 120 configurações para o algoritmo IBEA e 60 para o algoritmo NSGAI visto que este não utiliza população auxiliar. A tabela I apresenta o valores que foram variados nos experimentos.

TABELA I: Parâmetros utilizados nos experimentos

Parâmetro	Valores
População	200,250,300,350,400
Pop Aux. (IBEa)	200,300
Avaliações	40000,60000,70000,100000
Cruzamento	SinglePoint,TwoPoints,Uniform
Taxa Cruz.	90%
Mutação	BitFlip
Taxa Mut.	0.01%

Para fins de comparação foi utilizada uma cadeia de 64 aminoácidos, a cadeia mais complexa utilizada no estudo desenvolvido por Li et al. (2012) [2]. A fórmula abaixo representa a sequência de aminoácidos:



Por se tratarem de algoritmos estocásticos, cada configuração de algoritmo foi executado 30 vezes a fim

de avaliar se o comportamento se mantém similar em todas as execuções, resultando em 30 fronteiras ótimas referentes a cada execução de cada algoritmo. Para cada fronteira foram removidas as soluções duplicadas e dominadas. As fronteiras resultantes foram submetidas à normalização, dessa maneira, harmonizando as escalas dos objetivos utilizados pelos algoritmos. O cálculo utilizado para normalização é apresentado na equação 2.

$$X_{norm} = X - MIN/MAX - MIN \quad (2)$$

Onde X deve assumir os valores encontrados para um dos objetivos e MIN e MAX são o menor e o maior valor encontrados para o objetivo em questão. A normalização é necessária pois os conjuntos de valores para cada objetivo tem escalas diferentes. Ao normalizar garantimos que ambos os valores encontrados para cada objetivo estarão na mesma escala, de 0 e 1 neste caso. Dessa maneira ao calcular os valores de *hypervolume* evitamos que soluções com objetivos de escalas desproporcionais sejam priorizados.

Os valores de *hypervolume* foram calculados para cada uma das fronteiras geradas a partir das 30 execuções de cada configuração. Os valores de média, desvio padrão, mínimo e máximo referente as 30 execuções foram calculados para cada configuração. A tabela II apresenta os valores de *hypervolume* da melhor configuração obtida para cada algoritmo e o resultado do teste estatístico Kruskal-Wallis [19] entre as mesmas. Já a tabela III apresenta a melhor configuração para cada algoritmo.

TABELA II: Média, Desvio Padrão, Mínimo, Máximo e teste estatístico de Kruskal-Wallis

Algoritmo	Média	Desvio Padrão	Mínimo	Máximo	Kruskal-Wallis
NSGAI	0.5222	0.0429	0.4496	0.6025	TRUE
IBEA	0.5855	0.0711	0.4739	0.7799	

É possível observar baseado nas médias de *hypervolume*, com diferença estatística segundo o teste de Kruskal-Wallis, que o algoritmo IBEA obteve uma maior cobertura do espaço de busca quando comparado ao NSGAI. Dessa maneira podemos afirmar que o algoritmo IBEA obteve um desempenho melhor que o NSGAI pois a fronteira gerada pelo IBEA domina a fronteira gerada pelo NSGAI.

TABELA III: Melhores configurações de parâmetros para os algoritmos

Algoritmo	Avaliações	População	Pop Aux.	Cruzamento/%	Mutação/%
NSGAI	70000	350	-	TwoPoints/90%	BitFlip/0.01%
IBEA	100000	400	200	TwoPoints/90%	BitFlip/0.01%

Os melhores valores obtidos para o primeiro objetivo, de maximizar a quantidade de vizinhos topológicos, utilizando os algoritmos IBEA e NSGAI foram comparados com os resultados obtidos por métodos propostos em outros estudos que utilizaram a mesma sequência de aminoácidos, e são apresentados na tabela IV.

TABELA IV: Comparação entre a quantidade de vizinhos topológicos obtidos pelos estudos.

Metódo/Estudo	Menor Valor de Energia
MC [20]	35
GA [20]	37
SISPER [21]	39
ISA [22]	38
GAOSS [23]	42
GA-pull-moves [2]	42
IBEA	40
NSGAI	32

Analisando apenas o objetivo de maximizar a quantidade de vizinhos topológicos observamos que o IBEA conseguiu encontrar uma solução com 40, ficando atrás apenas de duas abordagens propostas por: Li et al. (2012) [2] e Huang et al. (2010) [23]. Já o algoritmo NSGAI ficou atrás de todas as abordagens pois obteve um valor muito baixo de vizinhos HH.

As figuras 4 e 5 apresentam respectivamente as soluções com a maior quantidade de vizinhos topológicos HH para o algoritmo IBEA e para o NSGAI.

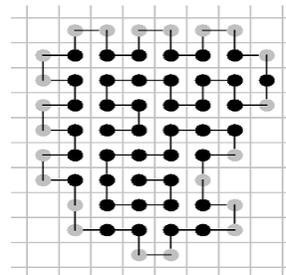


FIGURA 4: Melhor solução obtida pelo algoritmo IBEA com 40 vizinhos topológicos HH.

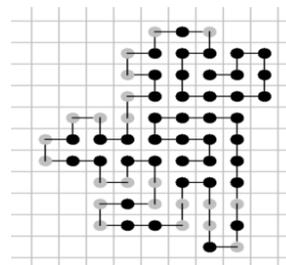


FIGURA 5: Melhor solução obtida pelo algoritmo NSGAI com 32 vizinhos topológicos HH.

VI. TRABALHOS CORRELATOS

Esta seção apresenta três trabalhos que possuem a implementação de estratégias para algoritmos evolutivos aplicados ao problema da PSP, e que serviram de base para a proposta desenvolvida para este trabalho.

Li et al. (2012) [2] desenvolveram um AE mono-objetivo combinado a uma estratégia de busca local baseada em *pull moves* para o modelo 2D-HP [24]. Li et al. (2012) realizaram experimentos utilizando cadeias de 20 a 64 resíduos

comparam os resultados obtidos com resultados de outros estudos e concluíram que abordagem proposta se mostrou eficiente quando comparada com outras estratégias utilizadas em estudos anteriores.

O estudo desenvolvido por Gabriel et al. (2012) [5] propõe um algoritmo evolutivo multi-objetivo para o problema PSP utilizando o modelo HP, o qual avalia a quantidade de contatos topológicos, assim como o grau de compactação da conformação. Concluíram que sua abordagem encontrou soluções com a mesma quantidade de contatos hidrofóbicos que um algoritmo mono-objetivo, utilizando populações menores do que as utilizadas em outras propostas e demonstrando um efeito positivo na utilização de uma proposta multi-objetiva.

Já Soares et al. (2011) [3] propuseram um AG multi-objetivo definindo o cálculo da energia de Van der Waals e o cálculo da energia eletrostática como os objetivos. Soares et al. (2011) realizaram experimentos com o algoritmo proposto e com o algoritmo NSGAI [4] e concluíram que o algoritmo proposto apresentou melhores resultados quando comparado ao NSGAI.

VII. CONCLUSÃO E TRABALHOS FUTUROS

A partir dos experimentos realizados, foi possível constatar que a aplicação de uma abordagem multiobjetiva para o problema PSP apresentou uma diferença entre os algoritmos IBEA e NSGAI. Verificamos uma diferença entre as médias de *hypervolume* entre os algoritmos. Contudo apenas ao verificar a maior quantidade de vizinhos topológicos encontrados por cada algoritmo ficou nítida a diferença no desempenho dos algoritmos. Ao comparar o algoritmo IBEA com outros estudos que realizaram experimentos com a mesma sequência de aminoácidos observamos que de 7 estudos: 5 obtiveram valores inferiores ao obtido algoritmo IBEA e apenas 2 estudos obtiveram valores superiores. Por fim também concluímos que operador de cruzamento que apresentou melhor desempenho foi o *TwoPoints* visto que foi aplicando este que os ambos os algoritmos NSGAI e IBEA obtiveram seus maiores resultados.

Trabalhos futuros incluem a exploração de mais técnicas de cruzamento, mutação, seleção; utilização de hiperheurísticas para seleção de operadores para os algoritmos; utilizar outras sequências de aminoácidos; introdução de outras técnicas de busca e adição de interface gráfica que viabilize interação entre usuários especialistas no domínio do problema com os MOEAs.

REFERÊNCIAS

- [1] H. S. Lopes, "Evolutionary algorithms for the protein folding problem: A review and current trends," in *Computational intelligence in biomedicine and bioinformatics*. Springer, 2008, pp. 297–315.
- [2] G. Li, J. Liu, Z. Liu, and Y. Zheng, "Genetic algorithm with pull moves for folding 2d model proteins," *International Journal of Digital Content Technology and its Applications*, vol. 6, no. 21, p. 412, 2012.
- [3] C. R. Soares Brasil, A. C. Botazzo Delbem, and D. R. Ferraz Bonetti, "Investigating relevant aspects of moeas for protein structures prediction," in *Proceedings of the 13th annual conference on Genetic and evolutionary computation*. ACM, 2011, pp. 705–712.
- [4] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *Evolutionary Computation, IEEE Transactions on*, vol. 6, no. 2, pp. 182–197, 2002.
- [5] P. H. Gabriel, V. V. de Melo, and A. C. Delbem, "Algoritmos evolutivos e modelo hp para predição de estruturas de proteínas," *Revista de Controle e Automação*, vol. 23, no. 1, pp. 25–37, 2012.
- [6] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *Parallel Problem Solving from Nature-PPSN VIII*. Springer, 2004, pp. 832–842.
- [7] M. Dorn, M. B. e Silva, L. S. Buriol, and L. C. Lamb, "Three-dimensional protein structure prediction: methods and computational strategies," *Computational biology and chemistry*, vol. 53, pp. 251–276, 2014.
- [8] K. F. Lau and K. A. Dill, "A lattice statistical mechanics model of the conformational and sequence spaces of proteins," *Macromolecules*, vol. 22, no. 10, pp. 3986–3997, 1989.
- [9] R. Santana, P. Larrañaga, and J. A. Lozano, "Protein folding in simplified models with estimation of distribution algorithms," *Evolutionary Computation, IEEE Transactions on*, vol. 12, no. 4, pp. 418–438, 2008.
- [10] M. A. C. Pacheco, "Algoritmos genéticos: princípios e aplicações," *ICA: Laboratório de Inteligência Computacional Aplicada. Departamento de Engenharia Elétrica. Pontifícia Universidade Católica do Rio de Janeiro. Fonte desconhecida*, 1999.
- [11] H. J. L. Baudrillard, *Manuel d'économie politique*. Guillaumin et cie, 1872.
- [12] C. M. Fonseca, J. D. Knowles, L. Thiele, and E. Zitzler, "A tutorial on the performance assessment of stochastic multiobjective optimizers," in *Third International Conference on Evolutionary Multi-Criterion Optimization (EMO 2005)*, vol. 216, 2005, p. 240.
- [13] E. M. d. N. Figueiredo, T. B. O. Ludermir, and C. J. A. C. Bastos Filho, "Algoritmo baseado em enxame de partículas para otimização de problemas com muitos objetivos," 2013.
- [14] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization," in *Genetic and Evolving Systems, 2008. GEFS 2008. 3rd International Workshop on*. IEEE, 2008, pp. 47–52.
- [15] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithms—a comparative case study," in *Parallel problem solving from nature—PPSN V*. Springer, 1998, pp. 292–301.
- [16] E. Zitzler, D. Brockhoff, and L. Thiele, "The hypervolume indicator revisited: On the design of pareto-compliant indicators via weighted integration," in *Evolutionary multi-criterion optimization*. Springer, 2007, pp. 862–876.
- [17] J. Durillo, A. Nebro, and E. Alba, "The jmetal framework for multi-objective optimization: Design and architecture," in *CEC 2010*, Barcelona, Spain, July 2010, pp. 4138–4325.
- [18] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. Da Fonseca, "Performance assessment of multiobjective optimizers: an analysis and review," *Evolutionary Computation, IEEE Transactions on*, vol. 7, no. 2, pp. 117–132, 2003.
- [19] W. H. Kruskal and W. A. Wallis, "Use of ranks in one-criterion variance analysis," *Journal of the American statistical Association*, vol. 47, no. 260, pp. 583–621, 1952.
- [20] R. Unger and J. Moult, "Genetic algorithms for protein folding simulations," *Journal of molecular biology*, vol. 231, no. 1, pp. 75–81, 1993.
- [21] J. L. Zhang and J. S. Liu, "A new sequential importance sampling method and its application to the two-dimensional hydrophobic-hydrophilic model," *The Journal of Chemical Physics*, vol. 117, no. 7, pp. 3492–3498, 2002.
- [22] W.-q. HUANG and H. Chen, "Improved simulated annealing algorithm for solving protein folding problem," *Application Research of Computers*, vol. 8, p. 021, 2005.
- [23] C. Huang, X. Yang, and Z. He, "Protein folding simulations of 2d hp model by the genetic algorithm based on optimal secondary structures," *Computational Biology and Chemistry*, vol. 34, no. 3, pp. 137–142, 2010.
- [24] N. Lesh, M. Mitzenmacher, and S. Whitesides, "A complete and effective move set for simplified protein folding," in *Proceedings of the seventh annual international conference on Research in computational molecular biology*. ACM, 2003, pp. 188–195.