UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

Algoritmos evolutivos e modelos simplificados de
proteínas para predição de estruturas terciárias

Paulo Henrique Ribeiro Gabriel

São Carlos – SP

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito: _____

Assinatura: _____

Algoritmos evolutivos e modelos simplificados de proteínas para predição de estruturas terciárias

Paulo Henrique Ribeiro Gabriel

Orientador: *Prof. Dr. Alexandre Cláudio Botazzo Delbem*

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação - ICMC-USP, como parte dos requisitos para obtenção do título de Mestre em Ciências - Ciências de Computação e Matemática Computacional.

USP - São Carlos
Fevereiro/2010

Agradecimentos

Ao finalizar este trabalho, quero expressar minha gratidão a todos aqueles que, direta ou indiretamente, contribuíram para a realização do mesmo.

Ao meu orientador, Prof. Dr. Alexandre Cláudio Botazzo Delbem, pela orientação oferecida, pelo incentivo e apoio durante toda a realização deste projeto.

Aos meus amigos, que tanto me auxiliaram neste período. Destaco aqui o apoio de Telma Woerle de Lima Soares, Daniel Rodrigo Ferraz Bonetti e Vinícius Veloso de Melo, por comentários e ideias, além de importantes ajudas com figuras e com a linguagem R. Também agradeço a Cristiane Regina Soares Brasil e Rodrigo Antônio Faccioli, por importantes sugestões.

Ao Programa de Pós-graduação de Ciências de Computação e Matemática Computacional (PG-CCMC) do Instituto de Ciências Matemáticas e de Computação (ICMC), pela oportunidade de realização do curso de mestrado. Aos funcionários e professores dispostos na prestação de serviços e colaboração. Destaco a imprescindível ajuda de Ana Paula Sampaio Fregona, da Seção Técnico Acadêmica, por sanar incontáveis dúvidas, e de Regina Célia Vidal Medeiros, da Seção de Atendimento ao Usuário da Biblioteca, por dicas de normatização deste documento. Incluo aqui, também, agradecimentos à Prof^a. Dr^a. Maria das Graças Volpe Nunes e ao Prof. Dr. Thiago Alexandre Salgueiro Pardo, por dicas e sugestões de escrita e organização do texto.

À Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) pela concessão da bolsa de mestrado e pelo apoio financeiro para a realização desta pesquisa.

E, finalmente, um agradecimento especial à minha família pelo apoio oferecido durante toda minha formação.

“It is our choices [...] that show what we truly are, far more than our abilities.”

J. K. ROWLING, *Harry Potter and The Chamber of Secrets* (1999)

Resumo

GABRIEL, P.H.R.. **Algoritmos evolutivos e modelos simplificados de proteínas para predição de estruturas terciárias**. 2010. 68 p. Dissertação (Mestrado) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2010.

A predição de estruturas de proteínas (*Protein Structure Prediction* — PSP) é um problema computacionalmente complexo. Para tratar esse problema, modelos simplificados de proteínas, como o Modelo HP, têm sido empregados para representar as conformações e Algoritmos Evolutivos (AEs) são utilizados na busca por soluções adequadas para PSP. Entretanto, abordagens utilizando AEs muitas vezes não tratam adequadamente as soluções geradas, prejudicando o desempenho da busca. Neste trabalho, é apresentada uma formulação multiobjetivo para PSP em Modelo HP, de modo a avaliar de forma mais robusta as conformações produzidas combinando uma avaliação baseada no número de contatos hidrofóbicos com a distância entre os monômeros. Foi adotado o Algoritmo Evolutivo Multiobjetivo em Tabelas (AEMT) a fim de otimizar essas métricas. O algoritmo pode adequadamente explorar o espaço de busca com pequeno número de indivíduos. Como consequência, o total de avaliações da função objetivo é significativamente reduzido, gerando um método para PSP utilizando Modelo HP mais rápido e robusto.

Palavras-chave: Predição de estrutura de proteínas, Algoritmos evolutivos, Otimização multiobjetivo, Modelo HP.

Abstract

GABRIEL, P.H.R.. Evolutionary algorithms and simplified models for tertiary protein structure prediction. 2010. 68 p. Dissertation (Master) – Institute of Mathematics and Computer Sciences, University of São Paulo, São Carlos, 2010.

Protein Structure Prediction (PSP) is a computationally complex problem. To overcome this drawback, simplified models of protein structures, such as the HP Model, together with Evolutionary Algorithms (EAs) have been investigated in order to find appropriate solutions for PSP. EAs with the HP Model have shown interesting results, however, they do not adequately evaluate potential solutions by using only the usual metric of hydrophobic contacts, hampering the performance of the algorithm. In this work, we present a multi-objective approach for PSP using HP Model that performs a better evaluation of the solutions by combining the evaluation based on the number of hydrophobic contacts with the distance among the hydrophobic amino acids. We employ a Multi-objective Evolutionary Algorithm based on Sub-population Tables (MEAT) to deal with these two metrics. MEAT can adequately explore the search space with relatively low number of individuals. As a consequence, the total assessments of the objective function is significantly reduced generating a method for PSP using HP Model that is faster and more robust.

Keywords: Protein structure prediction, Evolutionary algorithms, Multi-objective optimization, HP Model.

Lista de Figuras

2.1	Ilustração da estrutura de um aminoácido.	6
2.2	Ilustração do processo de formação de ligação peptídica.	8
2.3	Representação dos planos de ligação	8
2.4	Representação gráfica de uma α -hélice	9
2.5	Representação gráfica de uma folha- β	10
2.6	Representação em fitas da estrutura terciária de uma proteína	11
2.7	Exemplo de conformações em malha quadrática (2.7a) e em malha cúbica (2.7b) no Modelo HP.	14
3.1	Fluxograma de um AE típico.	19
3.2	Exemplos de representações e terminologia.	21
3.3	Representação gráfica do operador de mutação.	27
3.4	Exemplos de recombinação.	28
3.5	Exemplo de recombinação uniforme	28
3.6	Exemplo de inserção de indivíduos no AEMT com três subpopulações.	33
4.1	Exemplo de mutação em uma conformação de seis resíduos.	37
4.2	Dois exemplos de conformações para a sequência 273d.1	42
4.3	Número de avaliações da função objetivo pelo valor do <i>fitness</i> da sequência 643d.1.	44
4.4	Ilustração do processo de busca por soluções factíveis.	45
4.5	Exemplo de caso patológico em 2D.	46

Lista de Tabelas

2.1	α -aminoácidos e seus respectivos códigos de três letras e de uma letra.	7
3.1	Diferentes modelos de AEMO.	31
3.2	Diferenças fundamentais entre AEs.	33
4.1	Codificação utilizada por Unger e Moult (1993c).	38
5.1	Comparação dos resultados obtidos com sequências de 27 monômeros para o mono-objetivo.	48
5.2	Comparação dos resultados obtidos com sequências de 27 monômeros.	50
5.3	Comparação dos resultados obtidos com sequências de 64 monômeros.	51
5.4	Comparação do valor da aptidão na população inicial.	52
A.1	Sequências de 27 monômeros	67
A.2	Sequências de 64 monômeros	68

Lista de Siglas

ACO	<i>Ant Colony Optimization</i> (Otimização por Colônias de Formigas)
AE	Algoritmo Evolutivo
AEOM	Algoritmo Evolutivo para Otimização Multiobjetivo
AG	Algoritmo Genético
AEMT	Algoritmo Evolutivo Multiobjetivo em Tabela
DME	<i>Distance Matrix Error</i> (Erro na Matriz de Distâncias)
EES	Estratégias Evolutivas
IA	Inteligência Artificial
M_C	Matriz de Conformações
MEF	Máquina de Estados Finitos
MOGA	<i>Multiple Objective Genetic Algorithm</i>
NPGA	<i>Niched-Pareto Genetic Algorithm</i>
NSGA-II	<i>Elitist Non-Dominated Sorting Genetic</i>
PE	Programação Evolutiva
PESA	<i>Pareto Envelope-Base Selection Algorithm</i>
PSO	<i>Particle Swarm Optimization</i> (Otimização por Exame de Partículas)
PSP	<i>Protein Structure Prediction</i> (Predição de Estruturas de Proteínas)
RNM	Ressonância Nuclear Magnética
SPEA	<i>Strenght Pareto Evolutionary Algorithm</i>
VEGA	<i>Vector Evaluated Genetic Algorithm</i>

Sumário

Resumo	iii
Abstract	v
1 Introdução	1
1.1 Organização da dissertação	4
2 Predição de estrutura terciária de proteínas	5
2.1 Considerações iniciais	5
2.2 Composição e estrutura de proteínas	5
2.2.1 Estrutura secundária	9
2.2.2 Estruturas terciárias e quaternárias	10
2.3 O problema da predição da estrutura terciária	11
2.4 Modelos baseados em redes	13
2.5 Considerações finais	15
3 Algoritmos evolutivos	17
3.1 Considerações iniciais	17
3.2 Base biológica	18
3.2.1 O processo evolutivo	19
3.2.2 Terminologia básica	20
3.3 Algoritmos evolutivos canônicos	22
3.3.1 Programação evolutiva	23
3.3.2 Estratégias evolutivas	23
3.3.3 Algoritmos genéticos	24
3.4 Operadores de reprodução	26
3.4.1 Mutação	27
3.4.2 Recombinação	27
3.4.3 Operadores de reprodução para representação inteira	29
3.5 Algoritmos evolutivos para otimização multiobjetivo	29
3.5.1 Algoritmo evolutivo multiobjetivo em tabela	31

3.6	Considerações finais	32
4	Proposta de algoritmo evolutivo para predição de estruturas de proteínas	35
4.1	Considerações iniciais	35
4.2	Algoritmos evolutivos para predição de estruturas de proteínas	36
4.3	Abordagem proposta	40
4.3.1	Codificação das soluções	40
4.3.2	Funções objetivo	41
4.3.3	Operadores de reprodução	43
4.3.4	Tratamento de soluções infactíveis	43
4.4	Considerações finais	46
5	Experimentos e resultados	47
5.1	Considerações iniciais	47
5.2	Experimentos com AEs mono-objetivo	47
5.3	Experimentos com AEs multiobjetivo	49
5.3.1	Experimentos com sequências de 27 monômeros	49
5.3.2	Experimentos com sequências de 64 monômeros	50
5.3.3	Considerações sobre a utilização de M_C	52
5.4	Considerações finais	53
6	Conclusões	55
	Referências Bibliográficas	57
A	Sequências de <i>Benchmark</i> Utilizadas nos Experimentos	67

Introdução

Proteínas são polímeros de aminoácidos que executam uma grande variedade de funções no organismo. As enzimas, catalizadores específicos para diversas reações vitais de síntese e degradação, são exemplos de proteínas, assim como muitos hormônios reguladores. As proteínas também são componentes da membrana celular, funcionam como anticorpos, transportam oxigênio no sangue e fazem parte do material cromossômico. A forma, a regulação e a reprodução dos seres vivos são controladas pelas proteínas (Branden e Tooze, 1999; Lodish *et al.*, 2004).

Além disso, as proteínas têm importante papel na produção de medicamentos. Como a forma tridimensional de uma proteína está diretamente relacionada com a sua função no organismo (Setubal e Meidanis, 1997), a predição de estruturas de proteínas — do inglês *Protein Structure Prediction* (PSP) — passou a ser uma das áreas mais pesquisadas da Biologia Molecular. A estrutura de uma proteína pode ser determinada experimentalmente em laboratório por métodos de Cristalografia de Raios-X e Ressonância Nuclear Magnética (RNM) (Branden e Tooze, 1999). Esta última técnica é restrita a proteínas pequenas, enquanto que a primeira requer grande quantidade de experimentos com reduzida taxa de sucesso. Além disso, ambas demandam alto custo financeiro. Estima-se que a produção de um medicamento custe mais de US\$ 100 milhões, consumindo até dez anos de pesquisa, dos quais aproximadamente cinco correspondem à fase laboratorial (Blundell e Mizuguchi, 2000).

Por outro lado, a determinação da sequência de aminoácidos que compõem uma proteína pode ser considerada relativamente simples e pouco dispendiosa (Branden e Tooze, 1999). Por

esse motivo, tem-se investigado diversos algoritmos computacionais para resolução do problema de PSP a partir dessa sequência. Esses métodos são divididos em duas linhas (Ye, 2007): métodos baseados em conhecimento a priori e métodos de predição por primeiros princípios.

Os métodos baseados em conhecimento têm sido amplamente empregados, porém possuem uma série de limitações: dependem de grandes bancos de dados de estruturas de proteínas previamente determinadas, precisam de constante atualização desses bancos e de controle da redundância de informações, possuem problemas de alinhamento de sequências, necessitam de métodos computacionais eficientes que avaliem a similaridade entre estruturas e que determinem hierarquia de homologia estrutural (Jones, 2000; Marti-Renom *et al.*, 2000).

Por outro lado, predições por primeiros princípios não possuem tais restrições, dependendo unicamente da sequência de aminoácidos (Bonneau e Baker, 2001). Esses métodos baseiam-se no princípio de que essa sequência dobra-se em um estado em que sua **energia livre** é mínima. A energia de uma proteína (ou seja, da sequência dobrada) em função das posições dos seus átomos pode ser calculada utilizando, para tanto, modelos chamados de campos de força (Dill, 1985, 1999). A energia calculada a partir de um campo de forças pode ser minimizada e, dessa maneira, é possível encontrar qual é o estado de mínima energia que deve corresponder à forma tridimensional da proteína. Portanto, a solução deste problema pode ser mapeada em um **problema de otimização** de uma função que descreva a interação entre os átomos componentes das proteínas e desses com o solvente.

Tipicamente, um problema de otimização consiste em encontrar um conjunto

$$\vec{x} = (x_1, x_2, \dots, x_n) \in \mathcal{M}$$

de parâmetros de um dado sistema, de modo que uma **função objetivo** $f : \mathcal{M} \rightarrow \mathbb{R}$ seja minimizada, ou seja:

$$f(\vec{x}) \rightarrow \min \tag{1.1}$$

A solução do problema de otimização 1.1 consiste, assim, em encontrar um conjunto $\vec{x}^*$ tal que, $\forall \vec{x} \in \mathcal{M} : f(\vec{x}) \geq f(\vec{x}^*) = f^*$. Analogamente, problemas de maximização consistem em encontrar $\vec{x}$ tal que $\forall \vec{x} \in \mathcal{M} : f(\vec{x}) \leq f(\vec{x}^*) = f^*$. Como muitos problema de otimização do mundo real, a busca pela energia mínima de uma proteína apresenta complexidade computacional alta, de ordem exponencial (Setubal e Meidanis, 1997), não podendo ser solucionada em tempo viável por técnicas de força bruta (Berger e Leighton, 1998; Crescenzi *et al.*, 1998). A abordagem mais comum para esses problemas é a utilização de técnicas que encontram resultados aproximados, **sub-ótimos**. Entre essas técnicas, recebem destaque na literatura os **Algoritmos Evolutivos** (AEs) (De Jong, 2006).

Os AEs são ferramentas de otimização inspiradas na evolução natural e que têm sido aplicadas a muitos problemas nas mais diversas áreas do conhecimento humano (Coello Coello *et al.*, 2002; Deb, 2001; Eiben e Smith, 2003; Goldberg, 1989). Nos AEs é simulada a evolução de uma população de indivíduos que são possíveis soluções para o problema. Essas possíveis soluções evoluem de maneira que as mais adaptadas possuem maior probabilidade de sobreviver, isto é, tendem a estar presentes nas próximas populações.

Essa técnica apresenta como vantagem o fato de poder ser aplicada tanto sem utilizar informações específicas do problema (comportando-se como uma técnica genérica) como utilizando informações específicas de domínio, gerando algoritmos de aplicabilidade mais restrita, porém com maior desempenho (Deb, 2001).

Muitas pesquisas têm sido feitas no sentido de aplicar AEs para minimizar a energia livre das proteínas. Para modelar as soluções do problema (ou seja, as estruturas preditas), modelos simplificados têm sido comumente empregados. Um dos modelos mais bem estudado é o **Modelo Hidrofóbico-Hidrofílico**, conhecido como **Modelo HP** (Dill, 1985; Hart e Newman, 2006; Lau e Dill, 1989). Nesse modelo, classificado como modelo em redes, as interações hidrofóbicas são consideradas como sendo a principal força no processo de dobramento e a estrutura é representada em uma malha bi- ou tridimensional.

Diversos trabalhos têm aplicado AEs ao Modelo HP (Clote e Backofen, 2000; Custódio *et al.*, 2004; Johnson e Katikireddy, 2006; Krasnogor *et al.*, 1999; Patton *et al.*, 1995; Unger e Moulton, 1993c). Esses trabalhos visam apenas minimizar a função de energia simplificada do Modelo HP sem, contudo, considerar outras características das conformações geradas e, conseqüentemente, requerendo grande número de avaliações da função objetivo a fim de obter soluções apropriadas.

Este trabalho propõe uma formulação multiobjetivo para o problema de PSP em um Modelo HP. Essa abordagem considera o número total de interações hidrofóbicas e o grau de compactação das estruturas preditas, de modo a encontrar conformações compactas de energia livre mínima. A fim de avaliar eficientemente esses objetivos, foi utilizado um Algoritmo Evolutivo para Otimização Multiobjetivo (AEOM) (Deb, 2001) baseado em subpopulações, chamado **Algoritmo Evolutivo Multiobjetivo em Tabela (AEMT)**, cuja eficácia foi demonstrada em problemas combinatoriais multiobjetivos (Delbem, 2002; Delbem *et al.*, 2005; dos Santos, 2009; dos Santos *et al.*, 2008). Recentemente, um novo método (Ishibuchi *et al.*, 2009; Zhang e Li, 2007) com várias similaridades ao AEMT demonstrou a superioridade desse tipo de abordagem em relação aos AEOM baseados no conceito de dominância de Pareto. O diferencial do AEMT em relação à proposta apresentada por Ishibuchi *et al.* (2009) está na sua simplicidade de implementação (possuindo poucas alterações em relação ao AE padrão) e, principalmente, no fato de trabalhar com populações pequenas.

Experimentos computacionais com sequências de *benchmark* demonstraram a eficiência da adoção dessa abordagem multiobjetivo e do AEMT em relação a trabalhos anteriores. O tratamento mais adequado de múltiplos objetivos desse algoritmo, combinado à sua característica de populações pequenas, possibilita uma convergência relativamente rápida para o ótimo global do problema.

1.1 Organização da dissertação

Esta dissertação está organizada da seguinte forma:

- No **Capítulo 2** são apresentados conceitos sobre estruturas de proteínas e PSP. Neste capítulo também são descritos os modelos baseados em redes, como o Modelo HP.
- No **Capítulo 3** é descrito o funcionamento dos AEs, ferramentas utilizadas no desenvolvimento deste trabalho. Também são introduzidos conceitos de otimização multiobjetivo.
- O **Capítulo 4** apresenta a revisão bibliográfica de trabalhos envolvendo AEs e Modelo HP e descreve a metodologia proposta com base em tais trabalhos.
- No **Capítulo 5** são descritos os resultados obtidos com a implementação do AEOM para PSP, utilizando Modelo HP.
- Finalmente, no **Capítulo 6** apresenta-se as conclusões desse projeto de mestrado, caracterizando as contribuições e limitações do trabalho, bem como sugestões de desenvolvimentos futuros da abordagem proposta.

Predição de estrutura terciária de proteínas

2.1 Considerações iniciais

Este capítulo introduz conceitos relacionados ao problema de PSP. Inicialmente, descreve-se a composição de aminoácidos e proteínas e os níveis de dobramento de uma molécula proteica (Seção 2.2). Na Seção 2.3 é descrito o problema de PSP e na Seção 2.4 são apresentados os **modelos de redes** de PSP.

2.2 Composição e estrutura de proteínas

Uma proteína é uma macromolécula biológica composta por uma sequência de um alfabeto de 20 aminoácidos. Quimicamente, os elementos desse alfabeto são chamados **monômeros** e as proteínas (ou seja, as sequências de monômeros) são chamadas **polímeros**. As proteínas são responsáveis por diversas funções essenciais à manutenção da vida (Branden e Tooze, 1999). Com base em tais funções, as proteínas podem ser agrupadas em diversas classes (Alberts *et al.*, 2007; Lodish *et al.*, 2004):

Proteínas estruturais: fornecem suporte mecânico para células e tecidos;

Proteínas transportadoras: transportam pequenas moléculas ou íons;

Proteínas sinalizadoras: transportam sinais químicos de uma célula a outra;

Proteínas motoras: geram movimentos em células e tecidos;

Proteínas de armazenamento: armazenam pequenas moléculas ou íons;

Proteínas receptoras: detectam e transmitem sinais de modo a obter resposta celular;

Proteínas de regulação gênica: ligam-se ao DNA, ativando ou reprimindo a transcrição gênica.

Além dessas classes, existe um grupo de proteínas chamadas **enzimas**, capazes de catalisar uma grande quantidade de reações intra e extracelulares com velocidade e especificidade impossíveis de serem atingidas sem essas proteínas (Alberts *et al.*, 2007).

Cada aminoácido possui um grupo R (chamado **radical** ou **cadeia lateral**) que especifica o tipo do aminoácido. O átomo central de carbono é chamado **carbono- α** (C_α) e liga-se também a um **grupo amina** (NH_2), a um **grupo carboxila** ($COOH$) e a um átomo de hidrogênio (H). A Figura 2.1 ilustra a estrutura de um aminoácido.

Figura 2.1: Ilustração da estrutura de um aminoácido.

Na natureza são encontrados cerca de 300 diferentes cadeias laterais (Branden e Tooze, 1999). No entanto, um subconjunto de apenas 20 constituem as unidades de monômeros das quais as cadeias de proteínas são construídas. Nos organismos vivos, os 20 diferentes aminoácidos possuem propriedades químicas diferentes, especialmente variando sua hidrofobicidade, tamanho e carga elétrica, conforme mostrado na Tabela 2.1 (Lodish *et al.*, 2004; Polanski e Kimmel, 2007). Esses 20 aminoácidos são compostos orgânicos simples de baixo peso molecular, e são também chamados de **α -aminoácidos**.

A sequência de aminoácidos forma um encadeamento chamado **cadeia principal**, a qual é composta de átomos de carbono e nitrogênio interligados. O motivo $N - C_\alpha - C$ é chamado **unidade peptídica** e a ligação entre os átomos de carbono e nitrogênio recebe o nome de

Tabela 2.1: α -aminoácidos e seus respectivos códigos de três letras e de uma letra.

Aminoácido	Abreviação	Símbolo	Propriedades
Alanina	Ala	A	Não-polar, hidrofóbico
Cisteína	Cys	C	Polar, hidrofílico
Ácido Aspártico	Asp	D	Polar, hidrofílico
Ácido Glutâmico	Glu	E	Polar, hidrofílico
Fenilalanina	Phe	F	Não-polar, hidrofóbico
Glicina	Gly	G	Polar, hidrofílico
Histidina	His	H	Polar, hidrofílico
Isoleucina	Ile	I	Não-polar, hidrofóbico
Lisina	Lys	K	Polar, hidrofílico
Leucina	Leu	L	Não-polar, hidrofóbico
Metionina	Met	M	Não-polar, hidrofóbico
Asparagina	Asn	N	Polar, hidrofílico
Prolina	Pro	P	Não-polar, hidrofóbico
Glutamina	Gln	Q	Polar, hidrofílico
Arginina	Arg	R	Polar, hidrofílico
Serina	Ser	S	Polar, hidrofílico
Treonina	Thr	T	Polar, hidrofílico
Valina	Val	V	Não-polar, hidrofóbico
Triptofano	Trp	W	Não-polar, hidrofóbico
Tirosina	Tyr	Y	Polar, hidrofílico

ligação peptídica. Durante o processo de formação da ligação peptídica, ocorre liberação de uma molécula de água (H_2O); por esse motivo, os monômeros de uma cadeia proteica também são chamados **resíduos de aminoácidos** (ou simplesmente **resíduos**). A Figura 2.2 ilustra o processo de formação de ligações peptídicas e, conseqüentemente, de cadeias polipeptídicas.

A sequência de ligações peptídicas forma uma cadeia linear de resíduos que recebe o nome de **estrutura primária**. Em geral, uma cadeia curta de aminoácidos recebe o nome de **peptídeo** e as cadeias mais longas são os **polipeptídeo**. A direção da cadeia polipeptídica é determinada a partir do grupo amino terminal (N-terminal) até o grupo carboxila final (C-final ou C-terminal). É importante observar que a ligação peptídica é planar, ou seja, não existe uma rotação livre a redor da ligação. Conforme mostrado na Figura 2.3, no entanto, existe flexibilidade para rotação ao redor da ligação N – C_α (o chamado **ângulo diedral ϕ**) e ao redor da ligação C_α – C (**ângulo diedral ψ**). As proteínas dobram-se, portanto, em uma estrutura tridimensional porque o ângulo diedral ϕ pode rotacionar, assim como o ângulo ψ .

Figura 2.2: Ilustração do processo de formação de ligação peptídica.

Figura 2.3: Representação dos planos de ligação (Polanski e Kimmel, 2007). Em 2.3a, os átomos C e N formam uma ligação peptídica e os átomos C_α, H, O e C_α posicionam-se no mesmo plano. Em 2.3b, representação dos ângulos ϕ e ψ .

Como consequência das possíveis rotações de seus ângulo diedrais, a estrutura primária de uma proteína pode dobrar-se em uma estrutura tridimensional específica, gerando os demais níveis de organização estrutural da molécula proteica, descritos nas próximas seções.

2.2.1 Estrutura secundária

Diferentes regiões da sequência de aminoácidos tendem a formar regiões regulares, arranjos espaciais resultantes de dobramentos locais da cadeia, constituindo a chamada **estrutura secundária**. Um único polipeptídeo pode apresentar vários tipos de estrutura secundária, dependendo de sua sequência de aminoácidos. As duas principais categorias de estrutura secundária são estruturas periódicas, chamadas α -**hélice** e **folha- β pregueada**. Existe, também, ou uma volta sem estrutura definida, chamada “**random coil**”.

A α -hélice é uma estrutura em bastão. A cadeia principal forma a parte interna do bastão e as cadeias laterais posicionam-se para fora, conforme ilustrado na Figura 2.4. A hélice é estabilizada por pontes de hidrogênio entre os agrupamentos NH e CO da cadeia principal. Essas estruturas possuem 3,6 resíduos por volta e, em geral, têm um comprimento médio de 12 resíduos, o que corresponde a um comprimento de aproximadamente 18 Å.

Figura 2.4: Representação gráfica de uma α -hélice (Citizendium: The Citizens' Compendium, 2010). Em cima: representação em bastões, onde os átomos da cadeia principal (bastões pretos) formam uma volta, enquanto que os de grupos carboxila (vermelhos) formam pontes de hidrogênio com os do grupos amina (azuis). Embaixo: representação em fita (*ribbon*).

Em vez de fortemente enrolada, como a α -hélice, a folha- β pregueada é quase totalmente distendida. Outra diferença está no fato de que a estabilização por pontes de hidrogênio ocorrem em cadeias distintas, conforme ilustrado na Figura 2.5.

Estudos sugerem que determinados resíduos aparecem com mais frequência em hélices que em folhas, o que implica uma correlação entre tipos de aminoácidos e a estrutura secundária (Branden e Tooze, 1999). As estruturas secundárias são comumente encontradas combinadas

Figura 2.5: Representação gráfica de uma folha- β (Citizendium: The Citizens' Compendium, 2010). Em cima: vista lateral na representação em bastões e em fita. Em baixo: vista superior. Nota-se o posicionamento dos grupos carboxila (vermelhos) e amina (azuis) formando pontes. Observa-se, também, as posições oscilantes dos grupos carboxila (vermelho) e amina (azul) que formam ligações de hidrogênio entre as duas vertentes da folha.

em estruturas do tipo hélice-volta-hélice ou folha-volta-hélice, por exemplo. Essas unidades globulares são componentes das estruturas terciárias (Lodish *et al.*, 2004).

2.2.2 Estruturas terciárias e quaternárias

A **estrutura terciária** é a forma como a cadeia polipeptídica está enovelada. Esse nível estrutural ocorre quando domínios são dispostos entre si. A Figura 2.6 destaca os elementos de estrutura secundária presentes em uma molécula proteica. As estruturas terciárias também podem combinar-se em moléculas maiores, caracterizando a **estrutura quaternária**.

A conformação e a estabilidade da estrutura terciária é influenciada por um grande número de fatores, incluindo interações de van der Waals, força eletrostática, pontes de hidrogênio entre diferentes resíduos, pontes de hidrogênio com o solvente, além do efeito hidrofóbico (Branden e Tooze, 1999). Esses fatores permitem que uma sequência de aminoácidos seja dobrada em uma estrutura tridimensional, chamada **conformação** ou **estrutura nativa**. O processo de estabilização da estrutura primária em uma estrutura terciária recebe o nome de **dobramento** ou **enovelamento** proteico (Lodish *et al.*, 2004).

Figura 2.6: Representação em fitas da estrutura terciária de uma proteína e seus elementos de estrutura secundária (Citizendium: The Citizens' Compendium, 2010).

2.3 O problema da predição da estrutura terciária

Detalhes estruturais de uma molécula proteica estão relacionados diretamente com a função que a mesma executa no organismo. Muitos casos de patologia, por exemplo, são causados por proteínas defeituosas¹. Em geral, é possível determinar em laboratório a presença dessas proteínas em um organismo e sua função pode ser inferida pela interação com outras proteínas ou pela bioquímica do complexo ao qual pertence, mesmo que sua estrutura não tenha sido determinada (Lodish *et al.*, 2004). Por outro lado, a estrutura terciária da proteína também pode ser determinada experimentalmente por meio de dois métodos diferentes: Cristalografia de Raios-X e Ressonância Nuclear Magnética (RNM).

Na Cristalografia, a interação dos raios-X com os elétrons em moléculas arranjadas em um cristal é utilizada para obter um mapa de densidade de elétrons da molécula, o qual pode ser interpretado em termos de um modelo atômico (Branden e Tooze, 1999). A cristalização de proteínas pode ser difícil de obter e, geralmente, requer muitos experimentos diferentes variando um número de parâmetros, tais como pH, temperatura, concentração da proteína e a natureza do solvente. No método de RNM, as propriedades de *spin* magnético do núcleo atômico da molécula são utilizadas para obter uma lista das restrições de distância entre os átomos na molécula, a partir da qual a estrutura tridimensional da molécula da proteína pode ser obtida.

¹ A anemia falciforme é utilizada para ilustrar esse tipo de patologia (Amabis e Martho, 2006).

Este método não requer cristais de proteína e pode ser utilizado em moléculas proteicas em soluções concentradas. No entanto, sua utilização é restrita a pequenas moléculas de proteína (Branden e Tooze, 1999).

Observa-se, portanto, que ambas as técnicas requerem isolamento, purificação e cristalização das proteínas estudadas (Alberts *et al.*, 2007; Lodish *et al.*, 2004), o que pode dificultar ou até impossibilitar a determinação de estruturas. Além disso, essas técnicas são caras em termos de equipamentos e tempo gasto. Consequentemente, existem relativamente poucas proteínas cuja estrutura tridimensional é conhecida.

Por esse motivo, modelos computacionais de proteínas têm sido empregados de modo a tentar obter moléculas precisas com relação custo/tempo adequada. No entanto, o problema da predição de estrutura de proteínas — do inglês *Protein Structure Prediction* (PSP) — é um problema de otimização combinatorial, cuja solução está em um espaço de busca que cresce exponencialmente com o número de aminoácidos na proteína (Pal *et al.*, 2006; Setubal e Meidanis, 1997; Unger e Moult, 1993a). Por exemplo, sabe-se que os ângulos diedrais ϕ e ψ costumam assumir poucos valores independentemente uns dos outros e que, para cada par de valores, tem-se uma nova configuração da proteína (Setubal e Meidanis, 1997). Supondo que cada ângulo possa assumir apenas três valores, cada resíduo assumiria nove conformações distintas. Dessa forma, para uma proteína de apenas 100 resíduos existiriam 9^{100} possíveis configurações. Como as proteínas apresentam, em geral, milhares de resíduos e os ângulos de dobramento variam conforme a estrutura, é fácil observar que a determinação da estrutura proteica é um problema computacionalmente complexo, devido ao intratável número de combinações que a estrutura pode assumir.

Computacionalmente, duas abordagens têm sido exploradas (Ye, 2007): a **modelagem baseada em conhecimento** (Jones, 2000; Marti-Renom *et al.*, 2000) e a **modelagem por primeiros princípios**² (Bonneau e Baker, 2001).

Os métodos baseados em conhecimento são limitados pelo número de moléculas atualmente conhecidas. Além disso, a existência de proteína de estrutura bastante distinta das existentes em uma base de dados representa a grande limitação dessas técnicas. Por outro lado, nas abordagens por primeiros princípios, nenhuma similaridade na sequência de aminoácidos é necessária em relação às proteínas de estrutura desconhecida. A predição ocorre unicamente conhecendo-se a estrutura primária. Muitos desses métodos são baseados em otimização. Neste caso, dois problemas são comumente encontrados: (1) a especificação do modelo de otimização e (2) a escolha do algoritmo de busca. Entretanto, fatores como as interações entre os resíduos e as forças intramoleculares tornam o processo de PSP computacionalmente muito complexo,

²Também conhecida como modelagem (ou predição) *ab initio* ou *de novo*.

exigindo a criação de modelos computacionais muito elaborados, além da utilização de cálculos numéricos que estão além dos limites dos computadores atuais ou de um futuro próximo (Dill *et al.*, 2007).

Por estes motivos, muitas pesquisas focam o desenvolvimento de modelos simplificados de proteínas: abstrações matemáticas que ocultam muitos aspectos e enfatizam o efeito de outros (Chandru *et al.*, 2003; Hart e Newman, 2006). Análises e simulações computacionais de tais modelos podem ser comparados a resultados experimentais com moléculas reais, a fim de determinar se os aspectos enfatizados têm, de fato, o efeito esperado. Neste contexto, receberam destaque na literatura os **modelos em redes**, descritos na Seção 2.4.

2.4 Modelos baseados em redes

Os modelos baseados em redes (também chamados **modelos em lattice** ou simplesmente **modelos em redes**), são modelos simplificados de proteínas nos quais são aplicadas as seguintes simplificações (Backofen, 1999):

1. Os monômeros são representados por pontos posicionados nos vértices da malha;
2. As posições dos monômeros são restritas a posições em uma malha (ou *lattice*) regular;
3. O comprimento das ligações é único (ou restrito a poucas possibilidades);
4. A função de energia é simplificada.

O modelo em rede mais estudado é o chamado Modelo Hidrofóbico–Hidrofílico, ou Modelo HP, desenvolvido por Lau e Dill (1989). Neste modelo, o alfabeto dos 20 aminoácidos é reduzido a um alfabeto de apenas duas letras, H e P, onde H representa os aminoácidos **hidrofóbicos** e P representa os aminoácidos **polares** ou **hidrofílicos**. Esse modelo tem como base o fato de existir uma tendência de resíduos hidrofóbicos agruparem-se no interior da molécula proteica. O chamado **efeito hidrofóbico**, ou força hidrofóbica, é considerada a força predominante no processo de dobramento proteico (Dill, 1999; Yue e Dill, 1994), resultando na formação do chamado núcleo hidrofóbico.

Seguindo a notação adotada por Clote e Backofen (2000) e Backofen (1999), no Modelo HP uma sequência s de aminoácidos é um elemento em $\{H,P\}^*$. O i -ésimo elemento de s é denotado por s_i . Uma conformação c de uma sequência s de comprimento n (ou seja, uma sequência de n resíduos) passa a ser definida por:

$$c : [1..n] \rightarrow \mathcal{L}^d, \quad (2.1)$$

onde $\mathcal{L}^d$ é uma malha de dimensão $d = 2$ (para malhas bidimensionais ou quadráticas) ou $d = 3$ (no caso de malhas tridimensionais ou cúbicas). Além disso, duas condições são impostas:

1. $\forall 1 \leq i < n : \|c(i) - c(i+1)\| = 1$, onde $\|\cdot\|$ é a distância Euclidiana em $\mathcal{L}^d$;
2. $\forall i \neq j : c(i) \neq c(j)$.

A primeira condição é imposta pela estrutura da malha e determina que a distância entre dois resíduos consecutivos é 1 para qualquer configuração válida; a segunda condição evita a existência de ciclos e sobreposições (também chamadas colisões) de resíduos dentro da malha.

Em uma dada conformação c , existe ainda a distinção entre monômeros **conectados** e monômeros **vizinhos**. Dois monômeros i e j são ditos conectados em c se, e somente se, $j = i + 1$ ou $j = i - 1$. Deve-se observar que o número de monômeros conectados é fixo e independente da conformação c . Por outro lado, dois monômeros são vizinhos quando i e j não são conectados e $\|w(i) - w(j)\| = 1$. A Figura 2.7 ilustra um exemplo de conformação em $\mathcal{L}^2$ (Figura 2.7a) e em $\mathcal{L}^3$ (Figura 2.7b) com os diferentes tipos de monômeros³.

(a) Malha quadrática.

(b) Malha cúbica.

Figura 2.7: Exemplo de conformações em malha quadrática (2.7a) e em malha cúbica (2.7b) no Modelo HP.

No modelo HP, dada uma conformação válida, a **energia livre** E é dada pelo número negativo de vizinhos do tipo H-H. Assim, a conformação de energia mínima é a mesma que maximiza o número de contatos H-H (i.e., permite a obtenção de um núcleo compacto). Formalmente, o estado nativo de uma molécula em malha é dada por:

$$E = \sum_{1 \leq i < j \leq n} B_{i,j} \delta(r_i, r_j), \quad (2.2)$$

onde:

$$B_{i,j} = \begin{cases} 1 & \text{se os resíduos } i \text{ e } j \text{ são do tipo H;} \\ 0 & \text{caso contrário.} \end{cases}$$

³As cores utilizadas na Figura 2.7 para representar os resíduos hidrofóbicos (cinza) e polares (branco) seguem um padrão será mantido no restante deste texto, exceto quando explicitado o contrário.

$$\delta(r_i, r_j) = \begin{cases} 1 & \text{se os resíduos } i \text{ e } j \text{ são vizinhos;} \\ 0 & \text{caso contrário.} \end{cases}$$

Assim, pode-se concluir que para a conformação mostrada na Figura 2.7a o valor da energia livre é -2 .

Conceitualmente, o Modelo HP é relativamente simples se comparado com outras abordagens, além de permitir algumas extensões de modo a tornar-se mais elaborado (Backofen, 1999). Apesar disso, no Modelo HP, o problema de computar o estado nativo que maximiza o número de vizinhos H–H é tido como um problema $\mathcal{NP}$ -Completo tanto em duas dimensões, demonstrado formalmente por Crescenzi *et al.* (1998), quanto em três dimensões, como provado por Berger e Leighton (1998). Em ambos os casos, tal complexidade já era inferida, embora não completamente demonstrada por deferentes estudos (Fraenkel, 1993; Hart e Istrail, 1997; Paterson e Przytycka, 1996; Unger e Moulton, 1993a).

Por essa razão, diversos métodos baseados em heurísticas de otimização têm sido aplicados, incluindo Algoritmos Evolutivos (AEs) (Krasnogor *et al.*, 1998; Unger, 2004), Algoritmos Meméticos (Bazzoli e Tettamanzi, 2004; Krasnogor, 2002), algoritmos de Monte Carlo (Liang e Wong, 2001) e métodos de otimização por colônias de formigas — do inglês *Ant Colony Optimization* (ACO) (Shmygelska e Hoos, 2005) — e por exame de partículas — *Particle Swarm Optimization* (PSO) (Kanj *et al.*, 2009). As aplicações de AEs para PSP em modelos de redes são discutidas no Capítulo 4.

2.5 Considerações finais

Neste capítulo foram descritas algumas abordagens para o problema de PSP. Os métodos baseados em conhecimento dependem dos bancos de dados de estruturas, que precisam possuir uma proteína de estrutura conhecida com alta similaridade de sequência com a proteína cuja estrutura deve ser predita. Esse fato limita o número de estruturas que podem ser determinadas computacionalmente.

Por outro lado, os métodos de predição por primeiros princípios, que independem de qualquer homologia ou estrutura tridimensional, buscam prever a estrutura terciária de proteínas a partir somente da sequência de aminoácidos da proteína e dos conhecimentos sobre as interações bioquímicas. Esses métodos são importantes por estarem alcançando resultados relevantes, conseguindo prever estruturas de proteínas com relativa exatidão. Apesar dos avanços com essas técnicas, o universo de proteínas que se resolve com esses métodos ainda é restrito, uma vez

que consegue-se lidar apenas com moléculas relativamente pequenas por técnicas de primeiros princípios.

Uma alternativa para aumentar a capacidade para se modelar computacionalmente proteínas nativas tem sido o desenvolvimento de modelos de estruturas simplificados. Dentre esses modelos, os modelos de redes (como o Modelo HP) têm sido bastante investigados na literatura da área, aplicados em conjunto com AEs e outras técnicas de otimização.

Algoritmos evolutivos

3.1 Considerações iniciais

Algoritmos Evolutivos (AEs) são meta-heurísticas de otimização genéricas, baseadas em população, que utilizam mecanismos inspirados em Biologia, como mutação, recombinação, seleção natural e sobrevivência de indivíduos mais adaptados De Jong (2006); Eiben e Smith (2003). A mimetização em computador do processo de evolução natural é um eficiente e sistemático método de busca por valores ótimos no espaço de possíveis soluções de um problema de otimização.

A vantagem dessas técnicas em relação a outras técnicas de otimização está na possibilidade de modelar problemas pela simples descrição de uma potencial solução do mesmo. Isso possibilita que AEs possam ser facilmente adaptados para uma grande diversidade de problemas complexos. Em termos históricos, três abordagens de AEs foram desenvolvidas de forma independente (De Jong, 2006): a Programação Evolutiva (PE), as Estratégias Evolutivas (EEs) e os Algoritmos Genéticos (AGs). O princípio básico de todas essas técnicas é, no entanto, muito similar Eiben e Smith (2003): dada uma **população** de **indivíduos** (i.e., um conjunto de soluções), pressões do ambiente desencadeiam um processo de **seleção natural** (ou seja, um processo que privilegia em geral as melhores soluções até então encontradas), o que causa um incremento na taxa de adequação das soluções. Dada uma função a ser otimizada (maximizar ou minimizar), gera-se aleatoriamente um conjunto de soluções que são elementos pertencentes

ao domínio de uma função objetivo que, por sua vez, pode ser utilizada para avaliar a qualidade das soluções geradas. O valor resultante da avaliação da adequação é chamado **aptidão** (ou *fitness*).

Com base na aptidão, algumas das melhores soluções são selecionadas de modo a constituírem a uma nova **geração** pela aplicação de operadores de **recombinação** e/ou **mutação**. A recombinação (ou *crossover*) é um operador aplicado a duas ou mais soluções candidatas (chamadas **pais**) e resulta em duas ou mais novas soluções (**descendentes** ou **filhos**). A mutação é aplicada em uma candidata a fim de gerar outra. Ao final desse processo, as novas candidatas (descendentes) competem com as candidatas da geração anterior, com base no *fitness*, para assumir um lugar na nova geração. Esse processo é iterado até que uma candidata apresente uma solução que seja suficientemente qualificada ou até que um número máximo de iterações seja obtido.

Uma representação geral de um AE típico pode ser vista no fluxograma apresentado na Figura 3.1. É importante observar que vários componentes de um processo evolutivo são estocásticos: a seleção favorece indivíduos mais bem adaptados (ou seja, com melhor *fitness*), mas existe também a possibilidade de serem selecionados outros indivíduos. A recombinação dos indivíduos e a mutação são, em geral, aleatórias no sentido de que as alterações que promovem nas soluções seguem uma distribuição de probabilidades uniforme.

Vale destacar também que, diferente de outras técnicas de busca e otimização, as quais em geral constroem uma única solução por iteração¹, os AEs trabalham com conjuntos de soluções, o que tende a reduzir, em muitos casos, o número de iterações necessárias para a obtenção das soluções (em outras palavras, tende a reduzir o **tempo de convergência**).

Na Seção 3.2 serão abordados alguns aspectos básicos do processo evolutivo, base para o desenvolvimento de AEs, bem como a terminologia que será empregada no restante deste texto. Os principais tipos de AEs são descritos na Seção 3.3 e os operadores de reprodução são apresentados mais detalhadamente na Seção 3.4.

3.2 Base biológica

Os AEs podem ser vistos como técnicas de **Computação Bio-inspirada** (Floreano e Mattiussi, 2008), uma área de pesquisa que abrange uma série técnicas computacionais fundamentadas em conceitos biológicos. Assim, as técnicas evolutivas apresentam conceitos cuja origem está em diversos campos da Biologia, especialmente em ideias evolucionistas e na Genética. Esta seção foca nesses conceitos e sintetiza a terminologia empregada na definição de AEs.

¹Muitas dessas técnicas são descritas em (Sait e Youssef, 1999).

Figura 3.1: Fluxograma de um AE típico.

3.2.1 O processo evolutivo

Uma vez que os AEs são fortemente inspirados em processos evolutivos que ocorrem na natureza, é preciso explorar as bases de tais processos. Segundo De Jong (2006), os principais componentes dos sistemas evolutivos são:

1. Uma ou mais populações de indivíduos concorrendo por recursos limitados;
2. O conceito de aptidão, que reflete a habilidade do indivíduo para sobreviver e reproduzir-se;
3. A noção de mudanças dinâmicas nas populações devido ao nascimento e morte dos indivíduos;
4. O conceito de variabilidade pela hereditariedade, ou seja, os novos indivíduos possuem muitas das características de seus pais, embora não sejam idênticos a esses.

Tais conceitos foram inspirados na chamada **Teoria Sintética da Evolução**, também conhecida como **Neodarwinismo**. O Neodarwinismo admite que os principais fatores evolutivos são

a **mutação**, a **recombinação gênica** e a **seleção natural** (Amabis e Martho, 2006), resumidos a seguir.

Mutação gênica

A origem da variabilidade é a **mutação**, processo pelo qual o gene sofre alterações em sua estrutura, transmitindo-as ao se duplicar. Tais alterações são modificações na sequência de bases do DNA. Essa molécula, ao se autoduplicar, produzirá cópias idênticas de si, ou seja, diferentes da original, transmitindo hereditariamente a mudança. Isso pode acarretar na alteração da sequência de aminoácidos da proteína, modificando o metabolismo celular.

Recombinação gênica

O processo evolutivo seria lento se não fosse possível colocar juntas, em um mesmo indivíduo, mutações ocorridas em indivíduos da geração anterior. O fenômeno que possibilita esse evento é a **reprodução sexuada**. É importante considerar que a seleção natural não atua aceitando ou rejeitando mudanças individuais, mas sim escolhendo as melhores combinações gênicas entre todas variações presentes na população.

Seleção natural

A seleção natural é consequência de dois fatores:

1. Os membros de uma espécie diferem entre si;
2. A espécie produz descendência em maior número que aquele que de fato irá sobreviver.

Os indivíduos mais aptos a sobreviver são aqueles que, graças à variabilidade genética, herdaram a combinação gênica mais adaptada para determinadas condições naturais.

3.2.2 Terminologia básica

A seguir será apresentada a terminologia necessária (adaptada de (Sait e Youssef, 1999)) para o estudo de AEs.

Cromossomo, genes e alelos

A estrutura que codifica como os organismos são construídos é chamada **cromossomo**. Os cromossomos associam-se de modo a formar um organismo e seu número varia de uma espécie

para outra (Amabis e Martho, 2006). O conjunto completo de cromossomos de um ser vivo é chamado **genótipo** e as características do organismo gerado com base no genótipo constituem o **fenótipo**. De forma similar, a representação de soluções de um problema podem ser codificadas em uma estrutura de dados chamada cromossomo. Os cromossomos são codificados em um conjunto de símbolos chamados **genes**. Os diferentes valores de um gene são chamados **alelos**. A posição do gene em um cromossomo é denominada *locus* (Coello Coello *et al.*, 2002).

A representação das soluções candidatas (ou seja, dos indivíduos) é o primeiro estágio da elaboração de um AE e é crucial para o desempenho do algoritmo. Essa etapa consiste em definir o genótipo e a forma com que esse é mapeado no fenótipo. Dependendo da escolha, são necessários operadores de reprodução específicos (Seção 3.4).

A codificação mais simples é a **representação binária** que define o genótipo como um arranjo de 0s e 1s. É necessário definir o tamanho do arranjo, bem como o mapeamento genótipo–fenótipo. Entretanto, em muitas aplicações do mundo real, a representação binária pode apresentar fraco poder de expressão (Deb, 2001), não sendo eficiente na representação das possíveis soluções. Uma alternativa empregada é a **representação em ponto flutuante** ou **representação real**, segundo a qual as soluções são arranjos de números reais. Essa representação é usualmente empregada quando os genes são distribuídos em um intervalo contínuo, em vez de um conjunto de valores discretos (Eiben e Smith, 2003). A Figura 3.2 ilustra os termos discutidos nessa seção.

Figura 3.2: Exemplos de representações e terminologia.

Aptidão

O valor da aptidão (ou *fitness*) de um indivíduo (seja um genótipo ou um cromossomo) é um número positivo que mede o quão adequada é a solução. No caso de problemas de otimização, o *fitness* indica o custo da solução (se for um problema de minimização, as soluções de maior *fitness* são as de menor custo).

Pais, operadores de reprodução e descendentes

Os AEs trabalham sobre um ou mais cromossomos a fim de gerar novas soluções, chamadas **descendentes**. Os operadores que trabalham sobre cromossomos, chamados operadores de reprodução, são a **recombinação** (também chamada *crossover*) e a **mutação**. Esses operadores fazem analogia aos principais mecanismos de evolução natural, ou seja, a recombinação e a mutação gênica (Seção 3.2.1). A recombinação é aplicada a um par de cromossomos. Os dois indivíduos selecionados para o processo de recombinação são chamados **pais**. A mutação é aplicada a um simples cromossomo, modificando-o estocasticamente. Esses operadores serão descritos mais detalhadamente na Seção 3.4.

Geração e seleção

A **geração** é uma iteração do AE, na qual os indivíduos da população atual são selecionados e recombinações e/ou modificados, gerando descendentes. Devido à criação de novos descendentes, o tamanho da população cresce, sendo necessário um mecanismo de selecionar alguns indivíduos.

A ideia básica da **seleção** é a seguinte: seja uma população de tamanho M e seja N_d o número de descendentes, então, para a próxima geração, são selecionados M novos indivíduos entre as $M + N_d$ possíveis soluções, ou somente entre os N_d novos indivíduos, uma vez que N_d pode ser maior que M . Cada AE desenvolve, com base nesse princípio, uma estratégia de seleção (Seção 3.3).

3.3 Algoritmos evolutivos canônicos

Conforme mencionado, três diferentes abordagens evolutivas desenvolveram-se separadamente: Programação Evolutiva, Estratégias Evolutivas (EEs) e Algoritmos Genéticos. As principais características desses AEs (conhecidos como AEs canônicos) são importantes para o entendimento de propostas de solução baseadas em conceitos evolutivos, bem como para o desenvolvimento de novos AEs. As seções 3.3.1, 3.3.2 e 3.3.3 descrevem sucintamente os AEs canônicos.

3.3.1 Programação evolutiva

A Programação Evolutiva (PE) foi proposta por Fogel (1962) com o objetivo de utilizar os conceitos de evolução no desenvolvimento de Inteligência Artificial (IA).

Em PE, cada indivíduo da população é representado por uma Máquina de Estados Finitos (MEF), que processa uma sequência de símbolos. Durante a avaliação, os indivíduos são analisados por uma função de *payoff* de acordo com a saída da máquina e a saída esperada para solução do problema. A reprodução é feita apenas por operadores de mutação, sendo que todos os indivíduos da população atual geram novos descendentes. Esse processo caracteriza a chamada **reprodução assexuada**. Na seleção de indivíduos para a próxima geração, os descendentes competem com os μ pais e somente os indivíduos com maior *fitness* (no caso, os de maior *payoff* entre os $\mu + \lambda$ indivíduos) sobrevivem.

A PE garante que todos os indivíduos produzirão novos descendentes e somente os melhores indivíduos entre os atuais e os descendentes sobrevivem. O domínio total dos melhores indivíduos é chamado **elitismo total** (Kuri-Morales e Gutiérrez-García, 2001). O elitismo mais utilizado garante a sobrevivência apenas dos k -melhores indivíduos, $k < N$, onde N é o tamanho da população. O elitismo total, no entanto, pode diminuir significativamente a diversidade de indivíduos, podendo estagnar em ótimos locais e/ou aumentar o tempo de convergência do algoritmo (De Jong, 2006).

3.3.2 Estratégias evolutivas

As Estratégias Evolutivas (EEs) foram desenvolvidas com o objetivo de solucionar problemas de otimização de parâmetros. Foram propostas originalmente por Rechenberg (1965) e Schwefel (1975), na década de 1960, que desenvolveram a chamada (1 + 1)-EE em que um pai gera um único descendente (reprodução assexuada) e ambos competem pela sobrevivência.

Nas EEs, cada gene no cromossomo representa uma dimensão do problema, sendo que o alelo é representado em ponto flutuante. Os cromossomos são compostos por dois vetores, um com valores para cada dimensão e outro com o desvio padrão desses valores. A geração de um novo indivíduo é feita por meio da aplicação de um operador de mutação, com distribuição de probabilidade Gaussiana, com média zero e com desvio padrão do gene correspondente no pai. As EEs também utilizam elitismo completo.

O modelo original de EEs possui convergência lenta (De Jong, 2006). Por isso, foram desenvolvidos outros modelos denominados respectivamente (μ, λ) -EE e $(\mu + \lambda)$ -EE. No primeiro, μ pais morrem e sobrevivem λ descendente, sem competição entre os μ indivíduos já existentes

com os λ novos indivíduos. No segundo modelo, sobrevivem apenas μ indivíduos selecionados entre os μ indivíduos atuais e os λ novos indivíduos gerados.

3.3.3 Algoritmos genéticos

Os Algoritmos Genéticos (AGs) foram propostos por Holland (1962). A evolução natural é considerada um processo **robusto**, simples e poderoso, que poderia ser adaptado para obtenção de soluções computacionais eficientes para problemas de otimização. O conceito de robustez relaciona-se ao fato de os AGs, independentemente das escolhas dos parâmetros iniciais produzirem, em geral, soluções de qualidade (Goldberg, 1989). O principal diferencial dos AGs é a criação de descendente pelo operador de recombinação (De Jong, 2006).

Além disso, a utilização de operadores de mutação e recombinação equilibra dois objetivos aparentemente conflitantes: o **aproveitamento das melhores soluções** e a **exploração do espaço de busca**. O processo de busca é, portanto, multidimensional, preservando soluções candidatas e provocando a troca de informação entre as soluções exploradas (Michalewicz, 1996).

A seguir, são descritos os principais passos de um AG, baseando-se em Michalewicz (1996):

- Durante a iteração gen , um AG mantém uma população de soluções potenciais $P(gen) = \{x_1^{gen}, \dots, x_n^{gen}\}$;
- Cada indivíduo x_i^{gen} é avaliado produzindo uma medida de aptidão, ou *fitness*;
- Novos indivíduos são gerados a partir de μ indivíduos da população atual, os quais são selecionados para reprodução por um processo que tende a escolher indivíduos de maior *fitness*;
- Alguns indivíduos sofrem alterações, por meio de **recombinação** e **mutação**, formando novas soluções potenciais;
- Dentre as soluções antigas e novas ($\mu + \lambda$), são selecionados indivíduos (**sobreviventes**) para a próxima geração ($gen + 1$);
- Este processo é repetido até que uma condição de parada seja satisfeita. Essa condição pode ser um nível esperado de adequação das soluções ou um número máximo de iterações.

Nos AGs canônicos (Goldberg, 1989), as soluções candidatas são codificadas em vetores binários de tamanho fixo. A motivação para o uso de codificação binária é oriunda da **teoria dos**

esquemas (Holland, 1992) utilizada com sucesso para explicar por que os algoritmos genéticos funcionam. Holland (1992) argumenta que seria benéfico para o desempenho do algoritmo maximizar o paralelismo implícito inerente ao AG e apresenta uma análise qualitativa baseada no **Teorema dos Esquemas** sobre como um alfabeto binário poderia maximizar esse paralelismo. Entretanto, em diversas aplicações práticas a utilização de codificação binária leva a um desempenho insatisfatório. Em problemas de otimização numérica com parâmetros reais, algoritmos genéticos com representação inteira ou em ponto flutuante frequentemente apresentam desempenho superior à codificação binária (Deb, 2001; Michalewicz, 1996).

Segundo Michalewicz (1996) e Deb (2001), a representação binária apresenta desempenho pobre se aplicada a problemas numéricos com alta dimensionalidade ou se alta precisão numérica é requerida. Além disso, descrevem simulações computacionais comparando o desempenho de AGs com codificação binária e com ponto flutuante. Os resultados apresentados mostram uma clara superioridade da codificação em ponto flutuante para os problemas estudados.

Em contrapartida, Fogel (1994) argumenta que o espaço de busca por si só (sem levar em conta a escolha da representação) não determina a eficiência do AG. Espaços de busca de dimensão elevada podem às vezes ser explorados eficientemente, enquanto que espaços de busca de dimensão reduzida podem apresentar dificuldades significativas. Concorde, entretanto, que a maximização do paralelismo implícito nem sempre produz um desempenho ótimo.

O AG canônico utiliza um esquema de seleção de indivíduos para a próxima geração chamado **método da roleta** (Goldberg, 1989). Esse método é aplicado para selecionar λ indivíduos a partir de μ pais, mantidos em uma lista de reprodutores. Uma implementação comum deste algoritmo calcula uma lista de valores $[a_1, a_2, \dots, a_\mu]$, de modo que $a_i = \sum_{m=1}^i \mathcal{P}(m)$, onde $\mathcal{P}(m)$ é a probabilidade proporcional ao *fitness* de um indivíduo m passar para a próxima geração. O Algoritmo 1 mostra o pseudocódigo do método da roleta.

O desempenho de AGs pode, em muitos casos, ser melhorado forçando a escolha do melhor indivíduo encontrado em todas as gerações do algoritmo. Outra opção é simplesmente manter sempre o melhor indivíduo da geração atual na geração seguinte, estratégia essa conhecida como **seleção elitista** ou **elitismo** (Fogel, 1994; Michalewicz, 1996).

Outro exemplo de mecanismo de seleção é a seleção baseada em *rank* (Michalewicz, 1996). Esta estratégia utiliza as posições dos indivíduos quando ordenados de acordo com o *fitness* para determinar a probabilidade de seleção. Podem ser usados mapeamentos lineares ou não lineares para determinar a probabilidade de seleção. Uma forma de implementação desse mecanismo é simplesmente passar os n melhores indivíduos para a próxima geração.

Existe ainda a **seleção por torneio**, pela qual um subconjunto da população com k indivíduos é sorteado e os melhores indivíduos desse grupo são selecionados para decidir qual irá

Algoritmo 1: Pseudocódigo do método da roleta.

Entrada: μ , pais.
Saída: lista_reprodutores.
 $cont \leftarrow 1$;
 $a_i \leftarrow \sum_1^i \mathcal{P}(i)$;
enquanto $cont \leq \mu$ **faça**
 Obter um valor aleatório r de probabilidade uniforme em $[0,1]$;
 $i \leftarrow 1$;
 enquanto $a_i < r$ **faça**
 $i \leftarrow i + 1$;
 $a_i \leftarrow a_i + \mathcal{P}(i)$;
 fim
 lista_reprodutores[$cont$] $\leftarrow$ pais[i];
 $cont \leftarrow cont + 1$;
fim

reproduzir. Em geral, utiliza-se **torneio de 2** ($k = 2$), isto é, dois indivíduos (obtidos aleatoriamente da população) competem entre si e o ganhador (o de melhor *fitness*) torna-se um dos pais. O pseudocódigo do Algoritmo 2 ilustra esse procedimento para selecionar μ pais.

Algoritmo 2: Pseudocódigo do algoritmo de seleção por torneio.

Entrada: População, μ .
Saída: lista_reprodutores.
 $cont \leftarrow 1$;
enquanto $cont \leq \mu$ **faça**
 Obter k indivíduos aleatoriamente, com ou sem reposição ;
 Selecionar, com relação ao *fitness* o melhor indivíduo dentre k ;
 Denotar esse indivíduo por i ;
 lista_reprodutores[$cont$] $\leftarrow i$;
 $cont \leftarrow cont + 1$;
fim

Uma importante propriedade da seleção por torneio que esta não depende de um conhecimento global da população. Além disso, essa seleção não leva em consideração o *rank* que o indivíduo ocupa na população, permitindo uma seleção com menos tendências (Eiben e Smith, 2003).

3.4 Operadores de reprodução

Os operadores de reprodução, seguindo uma nomenclatura utilizada pela Biologia, podem ser divididos em dois grupos:

1. Operadores de reprodução **assexuada**, ou seja, um indivíduo sozinho gera um descendente, utilizando o operador de **mutação**;
2. Operadores de reprodução **sexuada**, quando é necessária a participação de pelo menos dois indivíduos na geração de descendentes, utilizando o operador de **recombinação**.

3.4.1 Mutação

O operador de mutação modifica aleatoriamente um ou mais genes de um cromossomo. Com esse operador, um indivíduo gera uma cópia de si mesmo, a qual pode sofrer alterações. A probabilidade de ocorrência de mutação em um gene é denominada **taxa de mutação**. Usualmente, são atribuídos valores pequenos para a taxa de mutação, uma vez que esse operador pode gerar um indivíduo potencialmente pior que o original.

Considerando a codificação binária, o operador de mutação padrão simplesmente troca o valor de um gene em um cromossomo (Goldberg, 1989). Assim, se o alelo de um gene selecionado é igual a 1, o seu valor passará a ser 0 após a aplicação da mutação (Figura 3.3)

Figura 3.3: Representação gráfica do operador de mutação.

No caso de problemas com codificação em ponto flutuante, os operadores de mutação mais populares são a **mutação uniforme** e a **mutação Gaussiana** (Deb, 2001).

3.4.2 Recombinação

O operador de recombinação é o mecanismo de obtenção de novos indivíduos pela troca ou combinação dos alelos de dois ou mais indivíduos. É o principal operador de reprodução dos AGs (Goldberg, 1989). Fragmentos das características de um indivíduo são trocadas por um fragmento equivalente oriundo de outro indivíduo. O resultado desta operação é um indivíduo que combina características potencialmente melhores dos pais.

Um tipo bastante comum de recombinação é a de 1-ponto. Nessa operação, seleciona-se aleatoriamente um ponto de corte nos cromossomos pais, dividindo cada pai em uma partição à direita e outra à esquerda do corte. Cada descendente é composto pela junção da partição à esquerda (direita) de um pai com a partição à direita (esquerda) do outro pai (Figura 3.4a).

Outra recombinação, de n -pontos, divide os cromossomos em n partições, as quais são recombinadas. A Figura 3.4b ilustra um exemplo com 2 pontos de corte. Neste caso, o Filho 1 (Filho 2) recebe a partição central do Pai 2 (Pai 1) e as partições à esquerda e à direita dos corte do Pai 1 (Pai 2).

Figura 3.4: Exemplos de recombinação.

Outro tipo de recombinação muito comum é a **recombinação uniforme**, que considera cada gene independentemente, escolhendo de qual pai o gene do filho será herdado. Em geral, cria-se uma lista de variáveis aleatórias (chamada **máscara**) de distribuição uniforme em $[0,1]$. Para cada posição, se o valor da variável aleatória for inferior a um dado $\mathcal{P}$ (usualmente 0,5), o gene será oriundo do Pais 1; caso contrário, virá do Pai 2. O segundo filho é gerado pelo mapeamento inverso. Alternativamente, pode-se gerar a máscara considerando-se apenas valores inteiros no intervalo $[0,1]$. Assim, se o valor da máscara for igual a 0, seleciona-se o gene do Pais 1. Caso contrário, seleciona-se um gene do Pai 2. (Figura 3.5).

Figura 3.5: Exemplo de recombinação uniforme. Nesse exemplo, utilizou-se a seguinte sequência de variáveis aleatórias $[0,1,0,0,0,1,0,1]$.

Como a recombinação uniforme troca genes individuais em vez de segmentos de genes, esse operador pode combinar características independentemente da sua posição no cromossomo. No entanto, De Jong (2006) destaca que não há nenhum operador de recombinação que claramente apresente um desempenho superior aos demais.

Em representações de ponto flutuante, foram propostos vários operadores que permitem a geração de novos alelos, além da simples troca de informação genética. Deb (2001) enumera alguns desses operadores.

3.4.3 Operadores de reprodução para representação inteira

Além da representação binária e em ponto flutuante, outras formas de codificação de AEs foram propostas (Eiben e Smith, 2003), como, por exemplo, a **representação inteira** onde os genes são elementos do conjunto dos números inteiro. Essa representação tem sido utilizada em problemas de busca de ótimos em um conjunto de inteiros. Nesse caso, torna-se mais eficiente modelar os cromossomos como vetores de inteiros, em vez de codificar as soluções em vetores binários.

A recombinação nesse tipo de representação pode ser realizada de forma análoga à utilizada em representação binária, ou seja, implementando-se operadores de recombinação de 1-ponto ou n -pontos, por exemplo (Seção 3.4.2). No caso da mutação, no entanto, houve a necessidade de desenvolver operadores específicos, como o *random resetting* e o *creep mutation* (Eiben e Smith, 2003). O operador *random resetting* atua de modo similar à mutação para codificação binária (Figura 3.3), substituindo um determinado gene por um valor escolhido aleatoriamente dentre os possíveis valores que os genes podem assumir. No caso do operador de *creep mutation*, soma-se um valor pequeno (positivo ou negativo) a valor de um determinado gene.

3.5 Algoritmos evolutivos para otimização multiobjetivo

Problemas de otimização multiobjetivo têm despertado grande interesse na área de Otimização. Nesses problemas, a qualidade da solução é definida com base na sua adequação em relação a diversos objetivos possivelmente conflitantes (Deb, 2001; Eiben e Smith, 2003). Na prática, os métodos de solução buscam reduzir esses problemas a outros com apenas um objetivo e depois buscam uma solução. Uma classe de métodos bastante utilizada nesse contexto é a dos **métodos baseados em pesos** ou simplesmente **método de pesos**, onde são atribuídos pesos às

diferentes funções objetivo, podendo ser realizado um processo de otimização da função de pesos (Arenales *et al.*, 2007).

Esses métodos, no entanto, são dependentes da escolha adequada dos pesos. Isso, em muitos casos, implica um conhecimento prévio dos intervalos correspondentes aos pesos mais adequados. Por essa razão, métodos que tentam encontrar soluções que apresentam um compromisso com os vários objetivos sem a utilização de pesos passaram a ser explorados (Eiben e Smith, 2003). Nesse caso, não existe somente uma solução para o problema, mas sim um conjunto de soluções ótimas, denominado conjunto de **Pareto ótimo** ou **fronteira de Pareto** (Deb, 2001).

O primeiro AEOM desenvolvido foi proposto por Schaffer (1985) e denominado *Vector Evaluated Genetic Algorithm* (VEGA). Este algoritmo é um AG modificado que avalia cada objetivo separadamente. No entanto, as soluções da fronteira obtidas, em geral, possuem baixa diversidade.

Goldberg (1989) criou um procedimento que ordena as soluções baseado no conceito de **dominância** que fornece um valor de aptidão para uma solução proporcional ao número de soluções que esta domina, i.e., não perde em nenhum objetivo e é melhor em pelo menos um. Com isso as soluções com maior número de dominados possuem maior aptidão e, assim, terão uma maior quantidade de cópias na lista de soluções. Com o objetivo de manter a diversidade das soluções, sugeriu a utilização de um método de compartilhamento que calcula o nicho de cada solução dentro da fronteira que a solução pertence. Com base nas ideias iniciais de Goldberg, foram propostos vários modelos de AEOMs.

A principal diferença entre os AEs tradicionais e os AEOMs é o operador de seleção, dado que a comparação entre duas soluções deve ser realizada de acordo com o conceito de dominância. Em algumas propostas, como *Multiple Objective Genetic Algorithm* (MOGA) (Fonseca e Fleming, 1993) e *Strength Pareto Evolutionary Algorithm* (SPEA) (Zitzler e Thiele, 1998), o valor de aptidão é proporcional à dominância da solução. Os outros métodos, como o *Niched-Pareto Genetic Algorithm* (NPGA) (Horn *et al.*, 1994), utilizam a dominância e não calculam um valor de aptidão.

Os modelos de AEOM são usualmente classificados em dois grupos (Deb, 2001; Eiben e Smith, 2003):

1. **Não elitistas:** compreendem os modelos que como o próprio nome indica, não utilizam nenhuma forma de elitismo nas suas interações;
2. **Elitistas:** compreendem os modelos que empregam alguma forma de elitismo. Por exemplo, o SPEA e o *Pareto Envelope-Base Selection Algorithm* (PESA) (Corne *et al.*, 2000) utilizam uma população externa para armazenar as soluções não dominadas encontradas

até o momento. Por outro lado, o *Elitist Non-Dominated Sorting Genetic* (NSGA-II) (Deb *et al.*, 2000) combina a população atual com a população gerada e preserva as melhores soluções de ambas.

Um estudo realizado por Zitzler *et al.* (2001) concluiu que o elitismo melhora as soluções encontradas por um modelo de AEOM. A Tabela 3.1 (adaptada de Ticona e Delbem (2008)) apresenta os principais modelos de AEOM e seus respectivos autores, bem como sua classificação em relação à presença ou não de elitismo.

Tabela 3.1: Diferentes modelos de AEMO.

Sigla	Nome do Modelo	Autores	Elitista
VEGA	<i>Vector Evaluated Genetic Algorithm</i>	Schaffer (1985)	Não
WBGa	<i>Weight Based Genetic Algorithm</i>	Hajela e Lin (1992)	Não
MOGA	<i>Multiple Objective Genetic Algorithm</i>	Fonseca e Fleming (1993)	Não
NSGA	<i>Non-dominated Sorting Genetic Algorithm</i>	Srinivas e Deb (1994)	Não
NPGA	<i>Niched-Pareto Genetic Algorithm</i>	Horn <i>et al.</i> (1994)	Não
PPES	<i>Predator-Prey Evolution Strategy</i>	Laumanns <i>et al.</i> (1998)	Não
REMOEA	<i>Rudolph's Elitist Multi-Objective Evolutionary Algorithm</i>	Rudolph (2001)	Sim
NSGA-II	<i>Elitist Non-Dominated Sorting Genetic Algorithm</i>	Deb <i>et al.</i> (2000)	Sim
SPEA, SPEA-2	<i>Strenght Pareto Evolutionary Algorithm</i>	Zitzler e Thiele (1998), Zitzler <i>et al.</i> (2001)	Sim
TGA	<i>Thermodynamical Genetic Algorithm</i>	Kita <i>et al.</i> (1996)	Sim
PAES	<i>Pareto-Archived Evolutionary Strategy</i>	Knowles e Corne (1999)	Sim
MONGA-I MONGA-II	<i>Multi-Objective Messy Genetic Algorithm</i>	Veldhuizen (1999)	Sim
PESA-I, PESA-II	<i>Pareto Envelope-Base Selection Algorithm</i>	Corne <i>et al.</i> (2000), Corne <i>et al.</i> (2001)	Sim

3.5.1 Algoritmo evolutivo multiobjetivo em tabela

Um estudo conduzido por Ishibuchi *et al.* (2009), demonstrou que AEOM que trabalham com dominância (como o NSGA-II e o SPEA) podem ter seu desempenho prejudicado em problemas com grande quantidade de objetivos. Nesses problemas, todos indivíduos podem deixar de dominar algum dos objetivos, o que torna o processo de seleção ineficiente. Como consequência, a diversidade populacional torna-se pequena.

Uma alternativa eficiente, conforme observam os autores, consiste em tratar um problema multiobjetivo como um conjunto de problemas mono-objetivo independentes (Zhang e Li, 2007).

Assim, cada objetivo é avaliado em relação a um peso diferente. Além disso, apenas um indivíduo é considerado em cada objetivo.

Trabalhos anteriores a Ishibuchi *et al.* (2009) já haviam chegando, empiricamente, a conclusões semelhantes (Delbem *et al.*, 2005; dos Santos, 2009; dos Santos *et al.*, 2008). Esses trabalhos utilizaram o chamado Algoritmo Evolutivo Multiobjetivo em Tabela (AEMT), proposto originalmente por (Delbem, 2002). Esse algoritmo trabalha com várias subpopulações em paralelo armazenadas em tabelas. Nesse algoritmo, os melhores indivíduos para cada característica do problema são armazenados em sua respectiva subpopulação.

O funcionamento básico do AEMT é descrito a seguir. Inicialmente, gera-se um conjunto de subpopulações para armazenar indivíduos avaliados pelas diferentes funções objetivo. Para cada função avaliada é construída uma nova subpopulação, de modo a otimizar cada uma das características avaliadas. O indivíduo selecionado para a reprodução pode ser proveniente de qualquer subpopulação avaliada. Essa estratégia de seleção aumenta a diversidade entre os indivíduos que reproduzem de forma que as características de um indivíduo de uma subpopulação possam migrar para as demais subpopulações (dos Santos, 2009). Como consequência, aumenta-se a possibilidade do algoritmo evitar ótimos locais, aproximando-se de soluções com avaliações próximas de um ótimo global na fronteira de Pareto-ótima.

Um parâmetro importante para o desempenho do algoritmo é o tamanho de cada subpopulação, que indica o número máximo de indivíduos que podem permanecer na subpopulação $SubPop_i$ de uma geração para outra. As soluções geradas pelo AEMT podem ser armazenadas ou descartadas, dependendo do grau de adaptação do indivíduo a cada objetivo do problema. No processo de seleção de sobreviventes, um novo indivíduo é acrescentado a uma subpopulação $SubPop_i$ se sua adequação ao objetivo de $SubPop_i$ for melhor que pelo menos um indivíduo da mesma. O mesmo indivíduo pode ser incluído em mais de uma tabela de acordo com esse critério de seleção. Como a população é estacionária, os novos indivíduos substituem os piores.

A Figura 3.6 ilustra o processo de inserção de indivíduos nas subpopulações. É importante observar que, diferente de outros AEOMs, como o NSGA-II e o SPEA, o AEMT gera apenas um único indivíduo a cada iteração.

3.6 Considerações finais

Neste capítulo foram abordados os Algoritmos Evolutivos (AEs), poderosas ferramentas aplicadas principalmente em problemas de otimização. Os AEs têm como base ideias evolucionistas, utilizando conceitos de Genética e dinâmica populacional.

Figura 3.6: Exemplo de inserção de indivíduos no AEMT com três subpopulações. Em 3.6a, o Indivíduo 1 é melhor que pelo menos um indivíduos das subpopulações 1 e 3. Em 3.6b, o Indivíduo 2 é melhor que pelo menos um indivíduo de cada uma das três subpopulações. Assim, tanto o Indivíduo 1 quanto o 2 o pior indivíduo de cada subpopulação.

Foram apresentados os três AEs canônicos, historicamente desenvolvidos de forma independente mas que mantêm ideias comuns. A Tabela 3.2 apresenta um breve resumo dos pontos em que os AEs diferem entre si.

Tabela 3.2: Diferenças fundamentais entre AEs.

Tipo de AE	Representação	Operador
PE	Ponto flutuante	Mutação e seleção ($\mu + \lambda$)
EEs	Ponto flutuante	Mutação e seleção (μ, λ) ou seleção ($\mu + \lambda$)
AG	Originalmente binário; Ponto flutuante em algumas aplicações	Mutação, recombinação e seleção

Também foram introduzidos brevemente os Algoritmos Evolutivos para Otimização Multi-objetivo (AEOMs), utilizados em problemas de otimização multiobjetivo. O Algoritmo Evolutivo Multiobjetivo em Tabela (AEMT), que será utilizado neste trabalho, foi descrito em maior nível de detalhes.

É importante destacar os AEs são processos heurísticos, de modo que podem não gerar, necessariamente, soluções ótimas. Além disso, muitos problemas de otimização encontrados na mundo real podem ser resolvidos por métodos exatos de forma eficiente. Por esse motivo, soluções baseadas em AEs devem ser utilizadas como um último recurso na resolução de problemas, ou seja, devem se aplicadas quando métodos de solução exatos sejam consideradas ineficientes.

Como PSP pode ser estudado como um problema de otimização complexo, para o qual não existem métodos exatos eficientes (conforme é discutido na Seção 2.3), os AEs têm sido aplicados a PSP com a obtenção de resultados relevantes na busca por soluções adequadas. Este trabalho investiga modelos de PSP e AEs buscando melhorar a qualidade das predições. O Capí-

tulo 4.3 discute aspectos de implementação adotados na literatura, como a escolha da representação dos cromossomos, modificações no cálculo da função objetivo e métodos de melhoria de desempenho dos algoritmos, e apresenta a abordagem multiobjetivo proposta e um método de construção de população inicial que corrige as soluções infactíveis geradas pela representação absoluta.

Proposta de algoritmo evolutivo para predição de estruturas de proteínas

4.1 Considerações iniciais

Conforme discutido no Capítulo 2, diversos métodos computacionais foram propostos com o objetivo de tratar problemas de PSP em modelos de redes, incluindo técnicas envolvendo AEs (Cotta, 2003; Custódio *et al.*, 2004; Gabriel e Delbem, 2009; Johnson e Katikireddy, 2006; Khimasia e Coveney, 1997; Krasnogor *et al.*, 1999; Lima *et al.*, 2009; Patton *et al.*, 1995; Scapin e Lopes, 2004; Unger e Moulton, 1993b,c; Zhao, 2008). Neste capítulo, são descritos, inicialmente, aspectos de implementação adotados nos trabalhos relacionados a esta dissertação (Seção 4.2). Esses trabalhos motivaram a abordagem proposta, apresentada na Seção 4.3, onde são discutidos aspectos referentes à codificação dos indivíduos (Seção 4.3.1), modelagem das funções objetivo (Seção 4.3.2) e operadores de reprodução (Seção 4.3.3), bem como a proposta de um algoritmo para tratamento de soluções ineficazes (Seção 4.3.4). A Seção 4.4 conclui este capítulo.

4.2 Algoritmos evolutivos para predição de estruturas de proteínas

Nos AEs, os indivíduos (ou seja, as estruturas geradas), podem ser representadas por meio de **coordenadas Cartesianas** (Unger e Moult, 1993c) ou, mais frequentemente, por **coordenadas internas** (Cotta, 2003; Patton *et al.*, 1995; Scapin e Lopes, 2004).

Utilizando coordenadas Cartesianas, cada vértice na malha é representado por um conjunto de coordenadas. Assim, por exemplo, uma sequência de n resíduos em um malha quadrática pode ser codificada por um vetor $[(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)]$, onde (x_i, y_i) é o vértice ocupado pelo i -ésimo resíduo (Ye, 2007). Em malhas cúbicas, são utilizados vetores com três coordenadas: (x_i, y_i, z_i) .

Por outro lado, nas representações envolvendo coordenadas internas, a localização de um aminoácido é especificada em relação à posição espacial do aminoácido anterior. Assim, uma conformação pode ser representada por uma sequência de **movimentos** a partir de um ponto definido aleatoriamente na malha. A representação por coordenadas internas é classificada em duas categorias (Krane e Raymer, 2003; Ye, 2007): **representação absoluta** e **representação relativa**.

Na representação absoluta, supondo uma sequência s de n resíduos, uma possível conformação (ou seja, um indivíduo) para essa sequência é gerada por um conjunto de $n - 1$ deslocamentos. Nos modelos de duas dimensões, as escolhas para os possíveis movimentos ficam restritas aos seguintes valores: **acima** (*up*, U), **abaixo** (*down*, D), **direita** (*right*, R) e **esquerda** (*left*, L), enquanto em três dimensões existem ainda as posições **frente** (*front*, F) e **atrás** (*back*, B). Por exemplo, a conformação representada na Figura 4.1a (adaptada de Ye (2007)) poderia ser descrita pelo cromossomo correspondente $c_{absoluta} = [R, U, R, D, R]$.

A representação absoluta, no entanto, permite o aparecimento de soluções infactíveis por meio de **movimentos de retorno**, ou seja, por permitir que um determinado movimento seja exatamente o oposto do movimento anterior como, por exemplo U seguido de D , L seguido de R , e assim por diante. De modo a diminuir o número de soluções infactíveis, foi proposta a chamada **representação relativa** que limita o total de possíveis movimentos para o resíduo seguinte. Nessa representação, o sistema de referência não é fixo e cada movimento é representado em relação à direção do movimento anterior. Por exemplo, no caso de malhas quadráticas, são permitidas apenas três direções: **em frente** (*forward*, F), **à direita** (*right-turn*, R) e **à esquerda** (*left-turn*, L), sendo que o primeiro movimento é sempre F que, usualmente, posiciona o segundo resíduo na posição $(1,0)$. No caso de malhas cúbicas, são permitidos ainda os mo-

vimentos **acima** (U) e **abaixo** (D). Considerando novamente a Figura 4.1a, tem-se a sequência de movimentos $c_{relativa} = [F, L, R, R, L]$

Figura 4.1: Exemplo de mutação em uma conformação de seis resíduos.

Um estudo realizado por Krasnogor *et al.* (1998) comparou ambas representações em relação à aplicação dos operadores genéticos. Novamente em relação à Figura 4.1, considerando a representação relativa, a Figura 4.1a pode ser descrita pela sequência $c_{relativa} = [F, L, R, R, L]$. Assim, uma mutação simples aplicada ao terceiro gene poderia gerar duas possíveis conformações: $c_{relativa}^1 = [F, R, F, R, L]$ e $c_{relativa}^2 = [F, L, L, R, L]$, ilustrados na Figura 4.1b e Figura 4.1c, respectivamente. No entanto, caso a estrutura seja descrita pela representação absoluta, $c_{absoluta} = [R, U, R, D, R]$, para produzir as mesmas novas conformações, é necessário modificar todos os pontos a partir da terceira posição: $c_{absoluta}^1 = [R, U, U, R, U]$ e $c_{absoluta}^2 = [R, U, L, U, L]$. Assim, conclui Krasnogor *et al.* (1998), a representação absoluta requer a utilização de mutação em múltiplos pontos, de modo a provocar mudanças que permitam um maior aproveitamento do espaço de busca. A representação relativa, por outro lado, requer apenas mutação simples, de um ponto.

É importante observar que ambos mecanismos não eliminam totalmente o aparecimento de soluções infactíveis, uma vez que essas podem ser geradas colidindo-se resíduos não adjacentes em outros pontos da malha. Nesse caso, uma das técnicas mais utilizadas consistem em aplicar uma **função de penalidade** (Bazzoli e Tettamanzi, 2004; Cotta, 2003; Patton *et al.*, 1995) às conformações com colisões, atribuindo um alto valor para a energia das mesmas. Desse modo, os métodos de busca tendem a desfavorecer soluções infactíveis, eliminando-as durante o processo de busca a fim de favorecer as conformações factíveis.

Com base nas coordenadas internas, diversos trabalhos envolvendo AEs em PSP foram propostos. Um dos pioneiros foi o AG híbrido proposto por Unger e Moult (1993c), para malhas

quadráticas e estendido posteriormente (Unger e Moult, 1993b) para malhas cúbicas. Esse AG incorpora um método de busca de Monte Carlos, e utiliza representação absoluta e codificação binária de três *bits* para descrever o cromossomo (Tabela 4.1).

Tabela 4.1: Codificação utilizada por Unger e Moult (1993c).

Binário	Inteiro	Translação	Código
000	0	Acima (<i>Up</i>)	<i>U</i>
001	1	Esquerda (<i>Left</i>)	<i>L</i>
010	2	Frente (<i>Front</i>)	<i>F</i>
011	3	Trás (<i>Back</i>)	<i>B</i>
100	4	Direita (<i>Right</i>)	<i>R</i>
101	5	Abaixo (<i>Down</i>)	<i>D</i>

Como o algoritmo híbrido de Unger e Moult (1993c) atribui -1 a cada par de resíduos classificado como vizinhos (Seção 2.4), as melhores soluções são as que atribuem menor valor para a função objetivo. O algoritmo cria uma população de soluções idênticas e aplica um conjunto de mutações, cujo efeito é limitado pela heurística de Monte Carlo, que verifica a factibilidade das soluções. Apenas soluções factíveis são mantidas na população, ou seja, cada indivíduo gerado tem a conformação avaliada e, em caso de infactibilidade, um novo indivíduo é gerado. Recombinação de 1-ponto (Seção 3.4.2) é utilizada, novamente rejeitando soluções infactíveis, e toda população é substituída, exceto o melhor indivíduo (ou seja, o AG proposto é elitista).

Unger e Moult (1993b) propuseram, também, um conjunto de 20 sequências de *benchmark* geradas aleatoriamente, 10 com 27 monômeros e 10 com 64 (Tabela A.1 e Tabela A.2, no Apêndice A). Os resultados obtidos demonstraram mais eficiência que um método baseado unicamente em Monte Carlo, em termo de número de avaliações da função objetivo.

Posteriormente, Patton *et al.* (1995) propuseram um AG não híbrido, com base no algoritmo padrão de Goldberg (1989). Utilizavam as funções do pacote GALOPPS (Goodman, 1996) para o desenvolvimento do AG. Os autores desenvolveram a representação relativa e o algoritmo mantém as conformações infactíveis na população, introduzindo uma função de penalidade que reduz o valor do *fitness* dos indivíduos infactíveis da seguinte forma: cada colisão subtrai -2 no valor do *fitness*, além de não computar o total de contatos H–H das colisões. Como consequência, existe uma tendência desses indivíduos não serem selecionados durante a criação de uma nova geração, uma vez que sua aptidão tende a ser inferior em relação aos outros indivíduos. Além disso, utilizaram codificação binária, recombinação de 1-ponto e 2-pontos, além de mutação por inversão de bits (Seção 3.4). Os resultados obtidos foram comparados aos de Unger

e Moult (1993b), obtendo em alguns casos soluções de menor energia e um número menor de avaliações da função objetivo para todos os casos testados.

Khimasia e Coveney (1997) também focaram no desenvolvimento de uma AG canônico (Seção 3.3.3) para o Modelo HP, sem utilizar rotinas de pacotes específicos. Utilizaram representação absoluta, cromossomo binário e operadores genéticos clássicos (mutação simples e recombinação de 1-ponto). As soluções ineficazes também eram penalizadas e os resultados foram comparados com Unger e Moult (1993b), tendo melhor desempenho para sequências de 27 resíduos.

Outro trabalho envolvendo AGs foi desenvolvido por Custódio *et al.* (2004), que utilizaram representação absoluta e exploraram o efeito de diferentes operadores de recombinação e mutação baseados em conhecimento específico do domínio de PSP. Esse tipo de algoritmo tem sido chamado de **hiper-heurística** (Burke *et al.*, 2003), de forma a destacar que são AEs que possuem conhecimento sobre o domínio do problema. Posteriormente, Custódio (2008) buscou manter a diversidade populacional como forma de realizar uma maior investigação do espaço de busca. Além disso, desenvolveu um mecanismo de **reparo** com o qual corrige as soluções ineficazes. Assim, se um movimento gera uma conformação inválida, novos movimentos são gerados aleatoriamente até que uma conformação válida seja encontrada. Caso não existam conformações válidas possíveis para uma determinada configuração, o valor do *fitness* do indivíduo é mantido constante e igual a 0, podendo ser eliminado durante o processo evolutivo.

Além disso, Custódio (2008) desenvolveu um método de substituição de indivíduos na população baseado no **erro na matriz de distâncias** — do inglês *Distance Matrix Error* (DME). Assim, seja X a matriz das distâncias de uma estrutura, onde $x_{i,j}$ é a distância do átomo i ao átomo j na estrutura, e Y a matriz das distâncias da segunda estrutura, onde $y_{i,j}$ é a distância do átomo i ao átomo j na segunda estrutura, a DME é definida como a média da diferença das matrizes de distância X e Y :

$$\sqrt{\frac{\sum_{i=1}^{n-1} \sum_{j=i+1}^n (x_{i,j} - y_{i,j})^2}{\frac{n(n-1)}{2}}},$$

onde n é o número total de átomos da estrutura. Quanto mais baixo for o valor do DME mais exata é a predição da estrutura terciária da proteína.

Dessa forma, quando dois indivíduos são selecionados, ambos são comparados em relação ao valor da DME e, caso sejam semelhantes estruturalmente, o de menor energia é mantido na população. Esse processo é chamado **substituição parental por crowding** (Michalewicz, 1996), e tem como objetivo aumentar a diversidade populacional. É importante observar que o cálculo da DME é computacionalmente caro se comparado ao cálculo da função objetivo. Esse

custo é justificado pelo fato desse trabalho focar o desenvolvimento de AEs para modelos mais complexos.

Outras estratégias para AG foram propostas, como a utilização de *backtracking* para correção de colisões. Neste sentido, Cotta (2003) utilizou mecanismos de *backtracking* em malhas quadráticas e Johnson e Katikireddy (2006) em malhas cúbicas, obtendo menor número de avaliações da função objetivo nos casos avaliados.

Outros trabalhos (Bazzoli e Tettamanzi, 2004; Krasnogor, 2002; Krasnogor *et al.*, 1999) exploraram mecanismos de busca local em AGs, desenvolvendo os chamados Algoritmos Meméticos (Moscato, 1989), verificando que tal metodologia tende a melhorar o desempenho dos AGs para modelos de redes. Além disso, Krasnogor *et al.* (1999) e Flores e Smith (2003) argumentam que o processo de busca pode ser otimizado se as soluções ineficazes forem mantidas na população, uma vez que isso aumenta a diversidade populacional. No entanto, tais soluções precisariam ser penalizadas.

4.3 Abordagem proposta

Esta dissertação descreve o desenvolvimento de um AEOM aplicado ao problema de PSP em um Modelo HP. No projeto deste algoritmo foram utilizadas ideias descritas em trabalhos relacionados, apresentados anteriormente, de modo a desenvolver um AE eficiente e robusto. As decisões tomadas durante o desenvolvimento do algoritmo estão relacionadas nas subseções seguintes (4.3.1, 4.3.2, 4.3.3 e 4.3.4). Os resultados obtidos são apresentados e discutidos no Capítulo 5.

4.3.1 Codificação das soluções

O algoritmo proposto utiliza representação absoluta para malhas cúbicas. Apesar de simples e de não evitar movimentos de retorno, esta codificação tem sido utilizada em diversos trabalhos recentes. Além disso, pode-se utilizar um procedimento simples para detectar colisões e decidir entre aceitá-las, porém penalizando a conformação gerada (Seção 4.3.2), ou corrigi-las, utilizando um mecanismo de correção adequado (Seção 4.3.4).

Também foi adotada a codificação inteira para representar os indivíduos de modo a propor operadores que modifiquem de forma mais acentuada as conformações obtidas (Bazzoli e Tettamanzi, 2004; Custódio, 2008).

4.3.2 Funções objetivo

Quando um AE é utilizado em PSP, é necessário escolher uma função adequada para o cálculo de aptidão dos indivíduos. Neste trabalho, após um estudo que será apresentado na Seção 5.2, três funções objetivo passaram a ser adotadas.

A primeira função busca maximizar no número de contatos do tipo H–H. Além disso, soluções inactíveis são permitidas na população, porém são penalizadas de modo que um menor *fitness* é associado a essas soluções (Cotta, 2003; Krasnogor *et al.*, 1999; Patton *et al.*, 1995). De acordo com esse método, o *fitness* f de uma dada conformação c é determinado pela Equação 4.1 (Bazzoli e Tettamanzi, 2004).

$$f(c) = n_H(c) - \rho \cdot n_C(c), \quad (4.1)$$

onde $n_H(c)$ é o número de contatos hidrofóbicos e $n_C(c)$ é o número de pontos da malha em que ocorrem colisões (i.e., posições ocupadas por dois ou mais monômeros). Por simplicidade, é atribuído o valor 1 ao termo ρ e a energia livre, nesse caso, é dada por $-f(c)$.

Um forma simples de reduzir o número de cálculos necessário para estabelecer o valor da aptidão de um indivíduo foi proposta neste trabalho: inicialmente, calcula-se o valor de $n_C(c)$ e então, caso esse valor seja diferente de 0 (ou seja, caso não existam colisões), calcula-se o valor de $n_H(c)$. Caso haja colisões (ou seja, caso $n_C(c)$ seja maior que 0), é atribuído 0 ao termo $n_H(c)$ (Gabriel e Delbem, 2009). Dessa forma, evita-se calcular a aptidão para todos os indivíduos durante parte do processo evolutivo, e esse passam a ser classificados em dois grupos: os de *fitness* negativo (inactíveis) e os de *fitness* positivo (factíveis).

É importante observar que a função $f(c)$ atribui um valor inteiro à aptidão da conformação c . Apesar de largamente adotado na literatura, esse método tem como limitação o fato de não identificar duas soluções distintas, porém com mesmo número de contatos H–H. Como consequência, uma conformação c_1 mais compacta, porém com o mesmo de contatos que c_2 , terá a mesma probabilidade de ser selecionada que c_2 , apesar de ser potencialmente melhor.

Por exemplo, na Figura 4.2¹, tem-se a representação gráfica de duas conformações distintas geradas por um AG para a sequência de *benchmark* 273d.1 (Tabela A.1). Ambas configurações possuem oito contatos hidrofóbicos, ou seja, um valor inferior ao encontrado na literatura (nove, no caso). Porém, adotando-se apenas a Equação 4.1 como forma de calcular a aptidão, torna-se impossível distinguir entre as duas conformações, apesar de terem sido construídas por

¹Esta figura foi gerada utilizando uma rotina escrita em R. As esferas vermelhas representam os resíduos hidrofóbicos e as pretas, os hidrofílicos.

seqüências de movimentos distintas. Portanto, a busca por uma melhor configuração pode ser interrompida em ótimos locais.

Figura 4.2: Dois exemplos de conformações para a sequência 273d.1

Isso motivou o acréscimo de novos objetivos ao Modelo HP tradicional, de modo a avaliar o grau de compactação das conformações geradas. Assim, utiliza-se neste trabalho uma função que calcula a distância Euclidiana entre todos os monômeros da malha. Em oposição ao número de iterações hidrofóbicas, esse valor (um número real positivo) deve ser minimizado. Dois valores distintos são obtidos neste caso: o primeiro, apresenta a maior distância $d_{max}(c)$ entre todos os resíduos presentes na malha; já o segundo traz a média das distâncias $d_{avg}(c)$ entre os resíduos.

Uma função de ponderação $\delta(c)$ é adotada para de modo a ponderar os diferentes objetivos considerados, conforme mostrado na Equação 4.2

$$\delta(c) = \frac{f(c)}{d(c)}, \quad (4.2)$$

onde $f(c)$ é dado pela Equação 4.1 e $d(c)$ é um valor escolhido entre $d_{max}(c)$ e $d_{avg}(c)$. Essa escolha é processada com base na taxa de variação desses valores entre duas gerações subseqüentes. Assim, seja $d_{max}^t(c)$ o valor de $d_{max}(c)$ na t -ésima geração, a taxa de variação de $d_{max}(c)$ é dada por $\frac{d_{max}^t(c)}{d_{max}^{t-1}(c)}$. Analogamente, a taxa de variação de $d_{avg}(c)$ é dada por $\frac{d_{avg}^t(c)}{d_{avg}^{t-1}(c)}$. Um vez processados esses cálculos, escolhe-se como valor de $d(c)$ o valor de distância que apresentar maior variação de uma geração para a outra. Dessa forma, evita-se, por exemplo, casos em que $d_{max}(c)$ não varia de uma geração para a outra, apesar da conformação c ser mais compacta uma vez que houve diminuição em outras distâncias consideradas no cálculo de $d_{avg}(c)$.

4.3.3 Operadores de reprodução

Os operadores de recombinação e de mutação foram escolhidos com base nos trabalhos de Krasnogor *et al.* (1998), Krasnogor *et al.* (1999), Flores e Smith (2003) e (Custódio, 2008), além de serem motivados por resultados preliminares.

Krasnogor *et al.* (1999) e Flores e Smith (2003) demonstram que um operador de recombinação de múltiplos (Seção 3.2.1) pontos provê uma melhor troca de informações entre os diferentes indivíduos gerados. Um método proposto para escolher os pontos onde ocorrerá a troca de informação genética consiste em dividir o cromossomo em relação ao comprimento $l(s)$ da sequência. Uma forma de realizar essa divisão é mostrada na Equação 4.3, proposta por (Custódio, 2008).

$$n = \text{int}\left(\frac{l(s)}{10}\right), \quad (4.3)$$

onde $\text{int}()$ é uma função que retorna o valor inteiro mais próximo ao resultado da divisão. Assim, por exemplo, as sequências de 27 monômeros serão divididas em três pontos e as de 64 em seis pontos.

O operador de mutação, por sua vez, é aplicado aos novos indivíduos gerados pelo operador de recombinação. Conforme observado por Krasnogor *et al.* (1998) e Krasnogor *et al.* (1999), a opção pela representação absoluta exige um mecanismo de mutação diferente do tradicionalmente utilizado em AEs. Dessa forma, em vez de utilizar mutação em um único ponto da cadeia, é necessário a utilização de mutações em diversos genes dos indivíduos gerados, de modo a produzir grandes modificações nas conformações. Neste projeto, alguns indivíduos resultantes da recombinação têm dois genes consecutivos modificados, a partir de um determinado ponto escolhido aleatoriamente. Isso impede, por exemplo, que a mutação afete um único gene, produzindo filhos com poucas informações novas. Além disso, a taxa de mutação deve ser relativamente maior que o adotado em outros problemas tratados por AGs.

4.3.4 Tratamento de soluções inactíveis

Conforme mostrado na Seção 4.3.2, usualmente utiliza-se uma função de penalidade que reduz a probabilidade de indivíduos inactíveis serem selecionados para as próximas gerações. Um exemplo de execução de um AG utilizando o método de penalidade proposto na Seção 4.3.2 é ilustrado na Figura 4.3 (Gabriel e Delbem, 2009), onde observa-se a evolução do *fitness* do melhor e do pior indivíduo, além do *fitness* médio da população. A sequência de *benchmark* utilizada é a 643d.1 (Tabela A.2).

É possível observar que a aptidão média da primeira geração é muito baixa, o que significa que todos os indivíduos apresentam colisões, ou seja, são inactíveis. Além disso, durante o pro-

Figura 4.3: Número de avaliações da função objetivo pelo valor do *fitness* da sequência 643d.1.

cesso evolutivo, as soluções infactíveis parecem convergir rapidamente para valores negativos, em oposição às soluções factíveis, cuja convergência é lenta. Essas observações motivaram o desenvolvimento de um método para corrigir colisões e, portanto, iniciar a busca de soluções apropriadas em regiões promissoras. Esse método utiliza uma matriz, chamada Matriz de Conformações (M_C).

Matriz de Conformações

O mecanismo de correção de soluções infactíveis utiliza uma matriz, chamada Matriz de Conformações (M_C), para decodificar os cromossomos, de modo a evitar soluções infactíveis e ao mesmo tempo avaliar o número de contatos H–H. Nesse método, cada gene indexa uma posição da matriz M_C , representando uma malha 3D. Caso a posição indexada já esteja ocupada, ou seja, caso já exista um resíduo na coordenada (x,y,z) de M_C , ocorre uma colisão. Para evitar a colisão, um novo valor é gerado e decodificado em uma nova posição espacial. A Figura 4.4 ilustra parte de uma conformação em duas dimensões.

Pode-se observar pela Figura 4.4 que, em caso de colisão, é verificado também o tipo do resíduo presente na posição, comparando-o com o resíduo anterior. Caso ambos os resíduos sejam do tipo H, significa que existe um contato H, o qual é contabilizado. Desse modo, em uma única varredura do cromossomo, corrige-se as soluções infactíveis e calcula-se o valor do

Figura 4.4: Ilustração do processo de busca por soluções factíveis: 4.4a mostra as potenciais colisões (setas tracejadas) e a nova posição (seta grossa); 4.4b mostra a conformação resultante e o contato H – H (linha tracejada).

aptidão. Para garantir correções em todo o cromossomo em apenas uma varredura, deve-se aplicar um procedimento eficiente. Assim, utiliza-se um procedimento de correção que é limitado por um número constante de passos por resíduo, conforme descrito a seguir.

Utiliza-se um vetor de permutações que consiste em um vetor com os possíveis deslocamentos (U, D, L, R, F, B) (Tabela 4.1), no caso codificados em valores inteiros. Para cada correção, gera-se uma permutação aleatória nesse vetor, de modo a evitar que a mesma sequência de movimentos seja aplicada em todo o cromossomo. Assim, quando se identifica uma colisão, cria-se uma nova permutação do vetor de deslocamentos e utiliza-se o valor contido na primeira posição do vetor como uma alternativa de movimento para eliminar a colisão. Caso esse movimento seja inválido, utiliza-se o movimento da próxima posição dentro do mesmo vetor, e assim por diante.

Caso todos os possíveis valores do vetor sejam testados sem que uma solução factível seja encontrada, o indivíduo é eliminado, ou seja, a construção da estrutura é cancelada e uma nova estrutura começa a ser gerada. Casos em que alterações de deslocamento locais não produzem factibilidade podem requerer rotinas de correções mais complexas, que consomem mais que um número constante de passos por colisão encontrada. Assim, em vez de corrigir o cromossomo, um novo cromossomo é construído nesse caso. A Figura 4.5 ilustra um caso patológico em 2D que não pode ser resolvido por meio de alterações locais. Para malhas em 3D, os casos patológicos que não podem ser resolvidos por alterações locais são mais raros (em experimentos, uma população de 100.000 indivíduos apresentou apenas 4 casos patológicos).

A reinicialização da matriz 3D para avaliação de um novo indivíduo requer um tempo cúbico em relação ao tamanho do cromossomo. Para evitar esse tempo, a cada indivíduo é atribuído um valor chamado “data de nascimento”, onde está armazenado um contador que indivíduos. Assim, o primeiro indivíduo gerado tem data de nascimento igual a 1, o segundo igual a 2 e

Figura 4.5: Exemplo de caso patológico em 2D: os três possíveis deslocamentos na matriz 2D correspondem a configurações infactíveis.

assim por diante. Com isso, o custo da análise de cada cromossomo é linear ao tamanho do cromossomo. Além disso, utiliza-se um espaço de memória cúbico para cada execução do AG. O tempo para alocação dessa memória é cúbico, porém é amortizado pelo total de indivíduos analisados sem necessidade de novas alocações ao longo das gerações do AG.

4.4 Considerações finais

Neste capítulo foi descrita a metodologia adotada no desenvolvimento de um AEOM para PSP. Para isso, foram apresentados os principais trabalhos envolvendo AEs e Modelo HP, de modo a dar embasamento para as decisões de projeto tomadas. No Capítulo 5 serão apresentados os resultados obtidos pelo algoritmo proposto.

Experimentos e resultados

5.1 Considerações iniciais

Seguindo a metodologia exposta no Capítulo 4.3, experimentos computacionais foram conduzidos de modo a validar a abordagem proposta. Inicialmente (Seção 5.2), são apresentados resultados preliminares que motivaram o desenvolvimento de um algoritmo multiobjetivo. A Seção 5.3 apresenta dos resultados do AEOM proposto, em sequência de 27 e 64 monômeros utilizadas como *benchmarks*. A Seção 5.4 conclui este capítulo.

5.2 Experimentos com AEs mono-objetivo

A adoção de uma estratégia multiobjetivo foi motivada por um conjunto de experimentos preliminares, que considerou duas implementações de um AG mono-objetivo. Na primeira implementação, $\mathcal{MO}1$, utilizou-se apenas o número de contatos hidrofóbicos como forma de avaliar os indivíduos (Equação 4.1). A segunda, $\mathcal{MO}2$, por outro lado, considerou apenas a ponderação entre o número de contatos H-H e a maior distância entre os monômeros dentro da malha (Equação 4.2).

Foram utilizadas as sequências de 27 resíduos, propostas por Unger e Moulton (1993b) e apresentadas na Tabela A.1. A taxa de mutação foi mantida constante em 3 % e a de recombinação

em 80 %. Para cada caso, o algoritmo foi executado 50 vezes. O AE foi implementado em C, padrão ISO/ANSI 1999, e nenhuma biblioteca auxiliar de AEs foi utilizada. Adotou-se as funções da biblioteca *GNU Scientific Library* (GSL)¹ para operações de geração de números aleatórios e permutações de vetores. Uma população de 120 indivíduos foi adotada.

A Tabela 5.1 compara os resultados obtidos por ambas implementações em relação ao número de contatos hidrofóbicos obtidos, ao número de avaliações da função objetivo — quando o algoritmo atinge valores ótimos — e à taxa de acertos, ou seja, o percentual de execuções nas quais o AG obteve a energia mínima. Em relação a $\mathcal{MO}2$, é apresentado, ainda o menor valor de $d(c)$ obtido e o valor da energia mínima correspondente a esse valor. O número de contatos H–H utilizados como comparação nos experimentos foram obtidos por Patton *et al.* (1995).

Tabela 5.1: Comparação dos resultados obtidos com sequências de 27 monômeros para o mono-objetivo.

Seq.	$\mathcal{MO}1$			$\mathcal{MO}2$			
	Aval.	Min. ($-f$)	Taxa de acertos	Aval.	Min. ($-f$)	Menor d	Taxa de Acertos
273d.1	205.096	−9	6	318.961	−9	1,46 (−8)	76
273d.2	–	−9	0	38.401	−10	1,83 (−10)	92
273d.3	34.858	−8	30	47.041	−8	2,41 (−8)	100
273d.4	–	−14	0	1.000.723	−15	3,21 (−14)	76
273d.5	36.458	−8	16	29.281	−8	0,80 (−8)	100
273d.6	160.378	−11	6	734.641	−11	2,13 (−10)	16
273d.7	905.604	−13	2	948.721	−13	2,91 (−12)	46
273d.8	13.176	−4	92	17.041	−4	0,74 (−4)	100
273d.9	12.314	−7	16	73.681	−7	0,86 (−7)	100
273d.10	10.164	−11	14	152.401	−11	1,38 (−11)	94
Média^a	24.137.804,80	−9,40	18,20	336.089,20	−9,60	1,77	80
Mediana	98.418	−9	10	113.041	−9,50	1,65	93

^aConsiderou-se no cálculo o total de avaliações da função objetivo realizadas pelo AE nos casos em que o ótimo não foi alcançado.

É possível observar que a utilização da função de ponderação $\delta(c)$ foi suficiente para aumentar, consideravelmente, a taxa de acertos do algoritmo apesar de, em alguns casos, não apresentar maior eficiência em relação ao número de avaliações da função objetivo. Pode-se observar também que o menor valor de $d(c)$ não leva, necessariamente, ao maior valor de $f(c)$. Isso leva à conclusão de que ambos objetivos (f e d) são independentes, apesar de d influenciar no comportamento de f em muitos casos. Dessa forma, buscou-se uma abordagem multiobjetivo, que otimize separadamente ambos valores e, portanto, permita que a combinação dos mesmo seja adequadamente explorada.

¹Disponível em: <http://www.gnu.org/software/gsl/>

5.3 Experimentos com AEs multiobjetivo

Com base nas conclusões apresentadas na Seção 5.2, foi proposta uma abordagem multiobjetivo para PSP em Modelo HP. Para otimizar os diferentes objetivos, foi utilizado o AEMT (Seção 3.5.1), para o qual foram criadas quatro subpopulações:

1. *SubPop1*: armazena os indivíduos avaliados em relação ao número de contatos hidrofóbicos. Esse objetivo deve ser maximizado;
2. *SubPop2*: armazena os indivíduos avaliados em relação à maior distância. Esse objetivo deve ser minimizado;
3. *SubPop3*: armazena os indivíduos avaliados em relação à média das distâncias. Esse objetivo deve ser também deve ser minimizado;
4. *SubPop4*: armazena os indivíduos avaliados em relação à função de ponderação apresentada na Equação 4.2. Esse objetivo deve ser maximizado.

Por convenção (Delbem *et al.*, 2005; dos Santos *et al.*, 2008), são utilizados tamanhos pequenos para as subpopulações, sendo que as de objetivos independentes, em geral, têm o mesmo número de indivíduos e a de ponderação é maior. Duas estratégias de geração de indivíduos forma adotadas: a primeira consiste em gerar as sequências de movimentos de forma aleatória, sem utilizar métodos de correção; a segunda, por outro lado, utiliza a matriz M_C para corrigir os indivíduos ineficazes da população inicial. O método de correção não foi adotado para os indivíduos de outras gerações, por provocar em alguns casos mudanças bruscas nas conformações geradas e por amenizar o efeito do operador de mutação, responsável pela variabilidade genética da população.

Nos experimentos foram utilizadas, além das sequências de 27 monômeros, sequências de 64 monômeros, também propostas por Unger e Moulton (1993b) e mostradas na Tabela A.2.

5.3.1 Experimentos com sequências de 27 monômeros

Para as sequências menores, utilizando-se AEMT sem M_C , avaliou-se apenas 8 indivíduos por geração, adotando-se os seguintes tamanhos de subpopulação: $SubPop1 = 1$, $SubPop2 = 2$, $SubPop3 = 2$ e $SubPop4 = 3$. No caso da utilização de M_C (AEMT+ M_C) foi necessário utilizar populações maiores, como será descrito posteriormente. Os resultados obtidos com o algoritmo multiobjetivo foram comparados com o trabalho de Patton *et al.* (1995) (P95), cujos

resultados eram melhores que os de Unger e Moulton (1993b), e com o trabalho de Johnson e Katikireddy (2006) ($\mathcal{J}06$), cujo mecanismo de *backtracking* apresentou resultados superiores aos de Patton *et al.* (1995) para as sequências menores (Tabela 5.2). Patton *et al.* (1995) e Johnson e Katikireddy (2006) utilizam recombinação de um e dois pontos, respectivamente, como probabilidade de 95 % e mutação de um gene, como probabilidade de 0,1 %.

Tabela 5.2: Comparação dos resultados obtidos com sequências de 27 monômeros.

Seq.	$\mathcal{P}95$		$\mathcal{J}08$		AEMT			AEMT+ M_C		
	Aval.	Min.	Aval.	Min.	Aval.	Min.	Acertos	Aval.	Min.	Acertos
273d.1	27.786	-9	15.854	-9	8.956	-9	80	318.961	-9	76
273d.2	81.900	-10	19.965	-10	12.891	-10	80	38.401	-10	92
273d.3	16.757	-8	7.991	-8	2.294	-8	100	47.041	-8	100
273d.4	85.447	-15	23.525	-15	19.837	-15	62	1.000.723	-15	76
273d.5	8.524	-8	3.561	-8	6.660	-8	76	29.281	-8	100
273d.6	44.053	-11	14.733	-11	38.317	-11	66	734.641	-11	16
273d.7	85.424	-13	23.112	-13	18.019	-13	72	948.721	-13	46
273d.8	3.603	-4	889	-4	1.848	-4	100	17.041	-4	100
273d.9	10.610	-7	5.418	-7	8.077	-7	100	73.681	-7	100
273d.10	16.282	-11	5.592	-11	11.006	-11	86	152.401	-11	94
Média	38.038,60	-9,60	12.064	9,60	12.790,50	-9,60	82,2	336.125,20	-9,60	70
Mediana	22.271,50	-9,50	11.362	9,50	9.981	-9,50	80	113.041	-9,50	84

É possível concluir que o algoritmo multiobjetivo foi capaz de encontrar o mesmo número de contatos hidrofóbicos que os outros algoritmos para todas as sequências menores. Ainda, é importante destacar que Patton *et al.* (1995) utilizam populações de 500 indivíduos enquanto Johnson e Katikireddy (2006) variam o tamanho da população entre 1000 e 1600 indivíduos. Nos experimentos apresentados na Tabela 5.2, utilizou-se apenas 8 indivíduos, o que demonstra o efeito positivo da adoção de múltiplos objetivos na busca por soluções mais adequadas. A taxa de acerto, ou seja, o total de execuções nas quais o algoritmo atingiu os valores ótimos, manteve-se superior a 60 % em todos os casos nos quais M_C não foi adotada.

5.3.2 Experimentos com sequências de 64 monômeros

No caso das sequências maiores, a comparação foi feita entre o AEMT e os trabalhos de Patton *et al.* (1995) ($\mathcal{P}95$) e Custódio (2008) ($\mathcal{C}08$), conforme mostrado na Tabela 5.3. Utilizou-se, aqui, os seguintes tamanhos de subpopulação: $SubPop1 = 1$, $SubPop2 = 1$, $SubPop3 = 1$ e $SubPop4 = 2$. A taxa de mutação adotada foi de 40 % e a de recombinação de 50 %.

Conclui-se que, no caso das sequências maiores, o algoritmo teve uma taxa de acertos relativamente baixa, não chegando a encontrar o ótimo no caso 643d.6. Porém, é importante destacar que o algoritmo proposto, quando obteve sucesso, atingiu um número de avaliações da função objetivo em muitos casos próximo aos valores reportados por outros autores. Merece

Tabela 5.3: Comparação dos resultados obtidos com sequências de 64 monômeros.

Seq.	$\mathcal{P}95$		$\mathcal{C}08$		AEMT		
	Aval.	Min.	Aval.	Min.	Aval.	Min.	Acertos
643d.1	433.533	−27	228.000	−31	654.320	−27	4
643d.2	167.017	−30	115.000	−36	1.117.558	−30	22
643d.3	172.192	−38	87.000	−44	244.936	−38	12
643d.4	107.143	−34	159.000	−39	2.374.943	−34	10
643d.5	154.168	−36	134.000	−40	2.266.601	−36	2
643d.6	454.727	−31	177.000	−33	–	−30	0
643d.7	320.396	−25	76.500	−28	644.915	−25	26
643d.8	315.036	−34	178.500	−36	2.646.709	−34	4
643d.9	151.705	−33	74.500	−38	623.314	−33	12
643d.10	191.019	−26	82.500	−31	139.571	−26	18
Média^a	246.693,60	−31,40	131.250	−35,60	13.071.286,70	31,30	11
Mediana	181.605,5	−32	124.750	−36	885.939	31,50	11

^aConsiderou-se no cálculo o total de avaliações da função objetivo realizadas pelo AE nos casos em que o ótimo não foi alcançado.

destaque, também, o pequeno número de indivíduos avaliados, ou seja, 5, em contraste com os 500 indivíduos avaliados por Patton *et al.* (1995) e Custódio (2008). Além disso, esse último utiliza substituição por *crowding*, operador de recombinação de múltiplos pontos e operadores de mutação específicos, o que é utilizado para justificar os bons resultados obtidos em todas as avaliações. Deve-se observar que a utilização de conhecimento adicional do domínio do problema é um aspecto positivo para a solução do problema de PSP (Bonneau e Baker, 2001). No entanto, isso muda a proposta original de somente utilizar como informação a sequência de aminoácidos para se determinar a estrutura de proteínas.

Posto esse contexto, pode-se dizer que o AEMT pode aproximar soluções de algoritmos com informação adicional sobre o problema mesmo sem utilizar tais informações. Acredita-se, ainda, que a inclusão de operadores como os propostos por Custódio (2008) no AEMT (e em outros AEs que não utilizem informações do domínio do problema) possa também aumentar seu desempenho.

É importante destacar que para as comparações em relação à energia mínima, considerou-se os valores obtidos por Patton *et al.* (1995). Também merece destaque o fato de, nos demais trabalhos, não ser apresentado o valor da taxa de acertos. Em experimentos que buscaram reproduzir os trabalhos de Patton *et al.* (1995), a taxa de acertos foi muito baixa, em alguns casos inferior a 1 %.

5.3.3 Considerações sobre a utilização de M_C

Com relação aos experimentos realizados, deve-se observar que os resultados envolvendo a Matriz de Conformações não podem ser considerados muito satisfatórios. Isso ocorre porque o mecanismo de correção de soluções infactíveis gera um conjunto de soluções cuja aptidão é baixa, ou seja, que possuem poucas (ou nenhuma) interações do tipo H–H, conforme mostrado na Tabela 5.4 (Gabriel e Delbem, 2009).

Tabela 5.4: Comparação do valor da aptidão na população inicial utilizando a Matriz de Conformações (M_C) e utilizando apenas representação absoluta (RA). Os valores são comparados aos ótimos definidos por Patton *et al.* (1995).

	Seq.	RA			M_C			ótimo
		melhor	média	pior	melhor	média	pior	
	273d.1	1	−7,03	−19	3	0,37	0	9
	273d.2	2	−7,02	−19	3	0,62	0	10
	273d.3	1	−7,03	−19	4	0,93	0	8
	273d.4	1	−7,80	−34	3	0,56	0	15
	273d.5	−1	−8,77	−22	3	0,55	0	8
	273d.6	1	−7,03	−19	5	1,10	0	11
	273d.7	0	−7,04	−19	2	0,16	0	13
	273d.8	0	−7,81	−34	3	0,35	0	4
	273d.9	0	−7,04	−19	3	0,30	0	7
	273d.10	0	−7,81	−34	3	0,75	0	11
Média		0,50	−7,44	−23,80	3,20	0,57	0	
	643d.1	−9	−23,52	−64	8	1,33	0	27
	643d.2	−9	−23,52	−64	5	1,32	0	30
	643d.3	−7	−24,69	−67	4	1,27	0	38
	643d.4	−7	−24,69	−67	7	1,77	0	34
	643d.5	−7	−23,50	−52	9	3,47	0	36
	643d.6	−7	−24,69	−67	8	2,14	0	31
	643d.7	−9	−23,52	−64	7	2,93	0	25
	643d.8	−7	−23,50	−52	6	1,86	0	34
	643d.9	−9	−23,52	−64	8	2,35	0	33
	643d.10	−7	−24,69	−67	7	1,60	0	26
Média		−7,80	−23,98	−62,80	6,90	2,00	0	

Assim, o AEMT+ M_C cria uma tendência de iniciar a busca em regiões de aptidão melhor que no caso de totalmente aleatório. Porém, essas soluções são provavelmente regiões promissoras que correspondem a soluções próximas de ótimos locais, prendendo a busca em regiões de sub-ótimos. Tal comportamento foi sugerido em outros trabalhos (Flores e Smith, 2003), mas pode ser amenizado pela utilização de populações iniciais grandes, contendo até 10.000 indiví-

duos. No entanto, a escolha de grande número de indivíduos resulta em um grande número de avaliações da função objetivo, uma vez que a população inicial precisa ser totalmente avaliada.

5.4 Considerações finais

Neste capítulo foram expostos os resultados do algoritmo multiobjetivo proposto neste trabalho de mestrado. Essa nova abordagem apresentou resultados satisfatórios quando comparada a outros trabalhos, abrindo a possibilidade de novos estudos que motivem a busca por um melhor desempenho da estratégia proposta.

Conclusões

Este capítulo resume as principais contribuições e limitações deste trabalho para a predição de estruturas de proteínas PSP em modelo HP. Além disso, são apresentados desenvolvimentos futuros que podem dar continuidade à abordagem proposta, superando algumas limitações e gerando outras contribuições.

O principal objetivo deste trabalho de mestrado foi o desenvolvimento de uma abordagem eficiente para PSP em Modelo HP. Isso motivou o estudo de algoritmos de computação evolutiva e sua aplicação em trabalhos relacionados. Nesse processo, buscou-se uma abordagem robusta que considere características de implementação que apresentaram bons resultados na literatura.

Assim, pôde-se observar a limitação do Modelo HP em relação ao tratamento de soluções sub-ótimas, uma vez que o modelo não considera que duas soluções com mesmo número de contatos hidrofóbicos possam ser distintas. Experimentalmente, demonstrou-se que a adoção de uma nova métrica, que calcula a compactação das estruturas preditas, apresenta resultados apropriados, aumentando consideravelmente a taxa de acertos do algoritmo.

O conhecimento adquirido com esses experimentos motivou a proposta e desenvolvimento de um Algoritmo Evolutivo para Otimização Multiobjetivo (AEOM). Foi utilizado o Algoritmo Evolutivo Multiobjetivo em Tabela (AEMT) que considera múltiplos objetivos e avalia, em geral, um pequeno número de indivíduos. As funções objetivo avaliadas pelo algoritmo consideram, simultaneamente, o número de contatos hidrofóbicos e a distância entre os monômeros das conformações geradas. Resultados obtidos pelo AEMT com sequências de *benchmark* peque-

nas apresentaram um número consideravelmente menor de avaliações da função de aptidão em relação a outros trabalhos encontrados na literatura. Além disso, o número de indivíduos avaliados é considerado pequeno. Resultados obtidos com as sequências maiores também podem ser considerados promissores.

Como forma de aumentar a robustez da técnica proposta, foi proposto um método de tratamento de soluções inválidas, de modo a buscar conformações adequadas com menor esforço computacional. Criou-se uma rotina que, auxiliada por uma matriz (chamada Matriz de Conformações (M_C)) e por um vetor de permutações, reconstrói eficiente as estruturas in-factíveis, gerando soluções factíveis. Resultados combinado esta estratégia com o AEMT, no entanto, não apresentaram melhora significativa no desempenho do algoritmo em relação ao número de avaliações do *fitness*, devido à grande quantidade de indivíduos necessários para a geração inicial.

Essa limitação e outras características desejáveis indicam os principais desenvolvimentos futuros a serem seguidos. Dentre eles destacam-se:

1. Adoção de métodos de controle de diversidade populacional, como os estudados por Tragante do Ó (2009), de modo que a busca por regiões promissoras não fique presa em ótimos locais quando se utiliza o AEMT+ M_C ;
2. Adoção de operadores de mutação específicos, como sugerido por Custódio (2008), de modo a realizar gerar novas conformações seguindo heurísticas mais adequadas;
3. Implementação de técnicas de busca local, cujo impacto positivo tem sido demonstrado em trabalhos relacionado (Bazzoli e Tettamanzi, 2004; Krasnogor, 2002);
4. Pesquisa do impacto de técnicas de aumento de desempenho de AEs desenvolvidas recentemente (Melo, 2009; Santana *et al.*, 2008).

Além disso, a adoção de técnicas de otimização multiobjetivo tem sido proposta (Cutello *et al.*, 2006; Lima, 2006; Tudela e Lopera, 2009) ou sugerida (Lima *et al.*, 2009) em trabalhos envolvendo modelos de proteínas mais complexos. Os resultados da implementação do AEMT motivam a proposta desse algoritmo para modelos mais elaborados e, portando, contribuir de forma mais significativa com o progresso na área de PSP.

Referências Bibliográficas

- ALBERTS, B.; BRAY, D.; LEWIS, J.; JOHNSON, A.; RAFF, M.; ROBERTS, K.; WALTER, P.; HOPKIN, K. *Fundamentos da biologia celular*. 2ªed. ARTMED Editora S.A., 864 p., 2007.
- AMABIS, J. M.; MARTHO, G. R. *Fundamentos da biologia moderna*, v. único. 4ªed. São Paulo, SP, Brasil: Editora Moderna Ltda., 839 p., 2006.
- ARENALES, M. N.; ARMENTANO, V. A.; MORABITO, R.; YANASSE, H. H. *Pesquisa operacional*. Rio de Janeiro, RJ, Brasil: Elsevier Editora Ltda., 523 p., 2007.
- BACKOFEN, R. *Optimization techniques for the protein structure prediction problem*. Tese de Doutorado, Ludwig-Maximilians-Universität, München, Institut für Informatik, München, Alemanha, 143 p., 1999.
- BAZZOLI, A.; TETTAMANZI, A. A memetic algorithm for protein structure prediction in a 3D-lattice HP model. In: RAIDL, G. R.; CAGNONI, S.; BRANKE, J.; CORNE, D.; DRECHSLER, R.; JIN, Y.; JOHNSON, C. G.; MACHADO, P.; MARCHIORI, E.; ROTHLAUF, F.; SMITH, G. D.; SQUILLERO, G., eds. *Applications of Evolutionary Computing, EvoWorkshops 2004: EvoBIO, EvoCOMNET, EvoHOT, EvoIASP, EvoMUSART, and EvoSTOC*, Coimbra, Portugal: Springer-Verlag, 2004, p. 1–10 (*Lecture Notes in Computer Science*, v.3005).
- BERGER, B.; LEIGHTON, T. Protein folding in the hydrophobic–hydrophilic (HP) model is NP-complete. *Journal of Computational Biology*, v. 5, n. 1, p. 27–40, 1998.
- BLUNDELL, T.; MIZUGUCHI, K. Structural genomics: an overview. *Progress in biophysics and molecular biology*, v. 73, p. 289–295, 2000.
- BONNEAU, R.; BAKER, D. Ab initio protein structure prediction: Progress and prospects. *Annual Review on Biophysics and Biomolecular Structures*, v. 30, p. 173–189, 2001.
- BRANDEN, C.; TOOZE, J. *Introduction to protein structure*. 2ªed. New York, NY, USA: Garland Publishing, Inc., 410 p., 1999.

- BURKE, E. K.; HART, E.; KENDALL, G.; NEWALL, J.; ROSS, P.; SCHULENBURG, S. Hyper-heuristics: An emerging direction in modern search technology. In: GLOVER, F.; KOCHENBERGER, G., eds. *Handbook of Metaheuristics*, Kluwer Academic, p. 457–474, 2003.
- CHANDRU, V.; DATTASHARMA, A.; ANIL KUMAR, V. S. The algorithmics of folding proteins on lattices. *Discrete Applied Mathematics*, v. 127, p. 145–161, 2003.
- CITIZENDIUM: THE CITIZENS' COMPENDIUM Protein structure. [On-line]. Disponível em http://en.citizendium.org/wiki/Protein_structure (Acessado em 16 fev. 2010)
- CLOTE, P.; BACKOFEN, R. *Computational molecular biology: An introduction*. San Francisco, CA, EUA: John Wiley & Sons, Inc., 300 p., 2000.
- COELLO COELLO, C. A.; VAN VELDHUIZEN, D. A.; LAMONT, G. B. *Evolutionary algorithms for solving multi-objective problems*. Genetic Algorithms and Evolutionary Computation. New York, NY, USA: Kluwer Academic, 576 p., 2002.
- CORNE, D. W.; JERRAM, N. R.; KNOWLES, J. D.; OATES, M. J. PESA-II: Region-based selection in evolutionary multiobjective optimization. In: SPECTOR, L.; GOODMAN, E.; WU, A.; LANGDON, W.; VOIGT, H.; GEN, M.; SEN, S.; DORIGO, M.; PEZESHK, S.; GARZON, M.; BURKE, E., eds. *GECCO 2001: Proceedings of the Genetic and Evolutionary Computation Conference*, Morgan Kaufmann Publishers, 2001, p. 283–290.
- CORNE, D. W.; KNOWLES, J. D.; OATES, M. J. The pareto envelope-based selection algorithm for multiobjective optimization. In: DEB, K.; G. RUDOLPH, X. Y.; LUTTON, E.; MERELO, J. J.; SCHWEFEL, H. P., eds. *Proceedings of the Parallel Problem Solving from Nature VI Conference*, Springer-Verlag, 2000, p. 839–848 (*Lecture Notes in Computer Science*, v.1917).
- COTTA, C. Protein structure prediction using evolutionary algorithms hybridized with backtracking. In: *IWANN '03: Proceedings of the 7th International Work-Conference on Artificial and Natural Neural Networks*, Berlin, Heidelberg: Springer-Verlag, 2003, p. 321–328.
- CRESCENZI, P.; GOLDMAN, D.; PAPADIMITRIOU, C.; PICCOLBONI, A.; YANNAKAKIS, M. On the complexity of protein folding. *Journal of Computational Biology*, v. 5, n. 3, p. 423–466, 1998.
- CUSTÓDIO, F. L. *Algoritmos genéticos para predição ab initio de estruturas de proteínas*. Tese de Doutorado, Laboratório Nacional de Computação Científica, Pretrópolis, RJ, Brasil, 198 p., 2008.
- CUSTÓDIO, 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*, v. 27, n. 4, p. 611–615, 2004.

- CUTELLO, V.; NARZISI, G.; NICOSIA, G. A multi-objective evolutionary approach to the protein structure prediction problem. *Journal of the Royal Society Interface*, v. 3, n. 6, p. 139–151, 2006.
- DE JONG, K. A. *Evolutionary computation: A unified approach*. Cambridge, MA, EUA: The MIT Press, 256 p., 2006.
- DEB, K. *Multi-objective optimization using evolutionary algorithms*. Wiley-Interscience Series in Systems and Optimization. New York, NY, EUA: John Wiley & Sons Inc., 515 p., 2001.
- DEB, K.; AGRAWAL, S.; PRATAB, A.; MEYARIVAN, T. *A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II*. KanGAL report 200001, Indian Institute of Technology, Kanpur, Índia, 11 p., 2000.
- DELBEM, A. C. B. *Restabelecimento de energia em sistemas de distribuição por algoritmo evolucionário associado a cadeias de grafos*. Tese de Doutorado, Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, Sp, Brasil, 140 p., 2002.
- DELBEM, A. C. B.; CARVALHO, A. C. P. L. F.; BRETAS, N. G. Main chain representation for evolutionary algorithms applied to distribution system reconfiguration. *IEEE Transactions on Power Systems*, v. 20, p. 425–436, 2005.
- DILL, K. A. Theory for the folding and stability of globular proteins. *Biochemistry*, v. 24, n. 6, p. 1501–1509, 1985.
- DILL, K. A. Polymer principles and protein folding. *Protein Science*, v. 8, p. 1166–1180, 1999.
- DILL, K. A.; OZKAN, S. B.; WEIKL, T. R.; CHODERA, J. D.; VOELZ, V. A. The protein folding problem: when will it be solved? *Current Opinion in Structural Biology*, v. 17, p. 342–346, 2007.
- EIBEN, A. . E.; SMITH, J. E. *Introduction to evolutionary computing*. Natural Computing Series. Berlin Heidelberg New York: Springer-Verlag, 299 p., 2003.
- FLOREANO, D.; MATTIUSSI, C. *Bio-inspired artificial intelligence: Theories, methods, and technologies*. Intelligent Robotics and Autonomous Agents. Cambridge, MA, USA: The MIT Press, 674 p., 2008.
- FLORES, S. D.; SMITH, J. Study of fitness landscapes for the HP model of protein structure prediction. In: *CEC 2003: Proceedings of the 2003 IEEE Congress on Evolutionary Computation*, Canberra, Austrália: IEEE Computer Society, 2003, p. 2338–2345.
- FOGEL, D. B. An introduction to simulated evolution. *IEEE Transaction on Neural Networks*, v. 5, n. 1, p. 3–14, 1994.
- FOGEL, L. Autonomus automata. *Industrial Research*, v. 4, n. 1, p. 14–19, 1962.

- FONSECA, C. M.; FLEMING, P. J. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In: FORREST, S., ed. *Proceedings of the Fifth International Conference on Genetic Algorithms*, University of Illinois at Urbana-Champaign, San Mateo, CA, EUA: Morgan Kauffman Publishers, 1993, p. 416–423.
- FRAENKEL, A. S. Complexity of protein folding. *Bulletin of Mathematical Biology*, v. 55, p. 1199–1210, 1993.
- GABRIEL, P. H. R.; DELBEM, A. C. B. Representations for evolutionary algorithms applied to protein structure prediction problem using HP model. In: GUIMARÃES, K. S.; PANCHENKO, A.; PRZYTICKA, T. M., eds. *Advances in Bioinformatics and Computational Biology (4th Brazilian Symposium on Bioinformatics, BSB 2009)*, Porto Alegre, RS, Brasil: Springer-Verlag, 2009, p. 97–108 (*Lecture Notes in Computer Science*, v.5676).
- GOLDBERG, D. E. *Genetic algorithms in search, optimization, and machine learning*. New York, NY, EUA: Addison-Wesley, 412 p., 1989.
- GOODMAN, E. D. *An introduction to GALOPPS: The “Genetic ALgorithm Optimized for Portability and Parallelism” systems*. Relatório Técnico 960701, Michigan State University, Michigan, EUA, 85 p., 1996.
- HAJELA, P.; LIN, C. Y. Genetic search strategies in multicriterion optimal design. *Structural Optimization*, v. 4, p. 99–107, 1992.
- HART, W. E.; ISTRAIL, S. Robust proofs of NP-hardness for protein folding: General lattices and energy potentials. *Journal of Computational Biology*, v. 4, n. 1, p. 1–22, 1997.
- HART, W. E.; NEWMAN, A. Protein structure prediction with lattice models. In: ALURU, S., ed. *Handbook of Molecular Biology*, Chapman & Hall/CRC Computer and Information Science Series, cap. 30, New York, NY, EUA: CRC Press, p. 30.1–30.24, 2006.
- HOLLAND, J. *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. Complex Adaptive Systems, 2ªed. Cambridge, MA, USA: The MIT Press, 211 p., 1992.
- HOLLAND, J. H. Outline for a logical theory of adaptive systems. *J. ACM*, v. 9, n. 3, p. 297–314, 1962.
- HORN, J.; NAFPLIOTIS, N.; GOLDBERG, D. E. A niched pareto genetic algorithm for multiobjective optimization. In: *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, Piscataway, NJ, EUA: IEEE Service Center, 1994, p. 82–87.
- ISHIBUCHI, H.; SAKANE, Y.; TSUKAMOTO, N.; NOJIMA, Y. Adaptation of scalarizing functions in MOEA/D: An adaptive scalarizing function-based multiobjective evolutionary algorithm. In: EHRGOTT, M.; FONSECA, C. M.; GANDIBLEUX, X.; HAO, J.-K.; SEVAUX, M., eds. *Evolutionary Multi-Criterion Optimization (Proceedings of 5th International Conference, EMO 2009)*, Nantes, França: Springer-Verlag, 2009, p. 438–452 (*Lecture Notes in Computer Science*, v.5467).

- JOHNSON, C. M.; KATIKIREDDY, A. A genetic algorithm with backtracking for protein structure prediction. In: KEIJZER, M.; CATTOLICO, M., eds. *GECCO 2006: Proceedings of the 8th Annual Conference on Genetic And Evolutionary Computation*, Washington, DC, EUA: ACM, 2006, p. 299–300.
- JONES, D. T. A practical guide to protein structure prediction. In: WEBSTER, D. M., ed. *Protein Structure Prediction: Methods and Protocol*, Methods in Molecular Biology, cap. 7, Totowa, NJ, USA: Humana Press, Inc, p. 131–154, 2000.
- KANJ, F.; MANSOUR, N.; KHACHFE, H.; ABU-KHZAM, F. Protein structure prediction in the 3D HP model. In: *ACS/IEEE International Conference on Computer Systems and Applications*, Los Alamitos, CA, EUA: IEEE Computer Society, 2009, p. 732–736.
- KHIMASIA, M. M.; COVENEY, P. Protein structure prediction as a hard optimization problem: The genetic algorithm approach. *Molecular Simulation*, v. 19, p. 205–226, 1997.
- KITA, H.; YABUMOTO, Y.; MORI, N.; NISHIKAWA, Y. Multi-objective optimization by means of the thermodynamical genetic algorithm. In: VOIGT, H.-M.; EBELING, W.; RECHENBERG, I.; SCHWEFEL, H.-P., eds. *PPSN IV: Parallel Problem Solving from Nature*, Berlin, Alemanha: Springer-Verlag, 1996, p. 504–512 (*Lecture Notes in Computer Science*, v.1141).
- KNOWLES, J. D.; CORNE, D. W. The pareto archived evolution strategy: A new baseline algorithm for multiobjective optimisation. In: *CEC 1999: Proceeding of the IEEE Congress on Evolutionary Computation*, Washington, DC, EUA: IEEE Service Center, 1999, p. 98–105.
- KRANE, D. E.; RAYMER, M. L. *Fundamental concepts of bioinformatics*. San Francisco, CA, EUA: Benjamin Cummings, 314 p., 2003.
- KRASNOGOR, N. *Studies on the theory and design space of memetic algorithms*. Tese de Doutorado, Faculty of Computing, Engineering and Mathematical Sciences, University of the West of England, Bristol, Inglaterra, 289 p., 2002.
- KRASNOGOR, N.; HART, W. E.; SMITH, J.; PELTA, D. A. Protein structure prediction with evolutionary algorithms. In: BANZHAF, W.; DAIDA, J.; EIBEN, A. E.; GARZON, M. H.; HONAVAR, V.; JAKIELA, M.; SMITH, R. E., eds. *GECCO 1999: Proceedings of the Genetic and Evolutionary Computation Conference*, Orlando, Flórida, EUA: Morgan Kaufmann, 1999, p. 1596–1601.
- KRASNOGOR, N.; PELTA, D.; LÓPEZ, P. M.; MOCCIOLA, P.; DE LA CANAL, E. Genetic algorithms for the protein folding problem: A critical view. In: ALPAYDIN, E.; FYFE, C., eds. *Proceedings of Engineering of Intelligent Systems*, Reading, MA, USA: Academic Press, 1998.
- KURI-MORALES, A. F.; GUTIÉRREZ-GARCÍA, J. Penalty functions methods for constrained optimization with genetic algorithms: A statistical analysis. In: COELLO COELLO, C. A.; ALBORNOZ, A.; SUCAR, L. E.; BATTISTUTTI, O. C., eds. *MICAI 2002: Proceedings of*

- the 2nd Mexican International Conference on Artificial Intelligence*, Springer-Verlag, 2001, p. 108–117 (*Lecture Notes in Artificial Intelligence*, v.2313).
- LAU, K. F.; DILL, K. A. A lattice statistical mechanics model of the conformational and sequence spaces of proteins. *Macromolecules*, v. 22, p. 3986–3997, 1989.
- LAUMANN, M.; RUDOLPH, G.; SCHWEFEL, H.-P. A spatial predator-prey approach to multi-objective optimization: A preliminary study. In: EIBEN, A. E.; SCHOENAUER, M.; SCHWEFEL, H.-P., eds. *PPSN V: Parallel Problem Solving From Nature*, Amsterdam, Holanda: Springer-Verlag, 1998, p. 241–249.
- LIANG, F.; WONG, W. H. Evolutionary monte carlo for protein folding simulations. *Chemical Physics*, v. 115, n. 7, p. 444–451, 2001.
- LIMA, T. W. *Algoritmos evolutivos para predição de estruturas de proteínas*. Dissertação de Mestrado, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, SP, 124 p., 2006.
- LIMA, T. W.; CALIRI, A.; DA SILVA, F. L. B.; TINÓS, R.; TRAVIESO, G.; DA SILVA, I. N.; DE SOUSA, P. S. L.; MARQUES, E.; DELBEM, A. C. B.; BONATTO, V.; FACCIOLI, R. A.; BRASIL, C. R. S.; GABRIEL, P. H. R.; TRAGANTE DO Ó, V.; BONETTI, D. R. F. Some modeling issues for protein structure prediction using evolutionary algorithms. In: DOS SANTOS, W. P., ed. *Evolutionary Computation*, cap. 9, Vukovar, Croácia: I-Tech, p. 153–178, 2009.
- LODISH, H.; BERK, A.; MATSUDAIRA, P.; KAISER, C. A.; KRIEGER, M.; SCOTT, M. P.; ZIPURSKY, L.; DARNELL, J. *Biologia celular e molecular*. 5ªed. Porto Alegre, RS, Brasil: ARTMED Editora S.A., 1054 p., 2004.
- MARTI-RENO, M. A.; STUART, A. C.; FISER, A.; SANCHEZ, R.; MELO, F.; ŠALI, A. Comparative protein structure modeling of genes and genomes. *Annual Review of Biophysics and Biomolecular Structure*, v. 29, p. 291–325, 2000.
- MELO, V. V. *Técnicas de aumento de eficiência para metaheurísticas aplicadas a otimização global contínua e discreta*. Tese de Doutorado, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, SP, 164 p., 2009.
- MICHALEWICZ, Z. *Genetic algorithms + data structures = evolution programs*. 3ªed. Berlin Heidelberg New York: Springer-Verlag, 382 p., 1996.
- MOSCATO, P. *On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms*. Relatório Técnico C3P 826, California Institute of Technology, Pasadena, CA, EUA, 67 p., 1989.
- TRAGANTE DO Ó, V. *Técnicas de controle de diversidade de populações em algoritmos genéticos para determinação de estruturas de proteínas*. Dissertação de Mestrado, Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto, Universidade de São Paulo, Ribeirão Preto, SP, Brasil, 97 p., 2009.

- PAL, S. K.; BANDYOPADHYAY, S.; RAY, S. S. Evolutionary computation in bioinformatics: a review. *IEEE Transactions on Systems, Man and Cybernetics—Part C: Applications and Reviews*, v. 36, n. 5, p. 601–615, 2006.
- PATERSON, M.; PRZYTICKA, T. On the complexity of string folding. *Discrete Applied Mathematics*, v. 71, p. 217–230, 1996.
- PATTON, A. L.; PUNCH III, W. F.; GOODMAN, E. D. A standard GA approach to native protein conformation prediction. In: *Proceedings of International Conference on Genetic Algorithms*, 1995, p. 574–581.
- POLANSKI, A.; KIMMEL, M. *Bioinformatics*. Berlin Heidelberg New York: Springer-Verlag, 376 p., 2007.
- RECHENBERG, I. *Cybernetic solution path of an experimental problem*. Relatório Técnico 1122, Royal Aircraft Establishment, Franborough, UK, 1965.
- RUDOLPH, G. Evolutionary search under partially ordered fitness sets. In: *ISI 2001: Proceedings of the International NAISO Congress on Information Science Innovations*, ICSC Academic Press: Millet/Sliedrecht, 2001, p. 818–822.
- SAIT, S. M.; YOUSSEF, H. *Iterative computer algorithms with applications in engineering: Solving combinatorial optimization problems*. Los Alamitos, CA, USA: IEEE Computer Society Press, 387 p., 1999.
- SANTANA, R.; LARRAÑAGA, P.; LOZANO, J. A. Protein folding in simplified models with estimation of distribution algorithms. *IEEE Transactions on Evolutionary Computation*, v. 12, n. 4, p. 418–438, 2008.
- DOS SANTOS, A. C. *Algoritmo evolutivo computacionalmente eficiente para reconfiguração de sistemas de distribuição*. Tese de Doutorado, Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, SP, Brasil, 166 p., 2009.
- DOS SANTOS, A. C.; DELBEM, A. C. B.; BRETAS, N. A multiobjective evolutionary algorithm with node-depth encoding for energy restoration. In: *ICNC '08: Fourth International Conference on Natural Computation*, 2008, p. 417–422.
- SCAPIN, M. P.; LOPES, H. S. Protein structure prediction using an enhanced genetic algorithm for the 2D HP model. In: *III Brazilian Workshop on Bioinformatics*, 2004, p. 183–186.
- SCHAFFER, J. D. Multiple objective optimization with vector evaluated genetic algorithms. In: *Genetic Algorithms and their Applications: Proceedings of the First International Conference on Genetic Algorithms*, Lawrence Erlbaum, 1985, p. 93–100.
- SCHWEFEL, H.-P. *Evolutionsstrategie und numerische optimierung*. Tese de Doutorado, Technical University of Berlin, Berlin, Germany, 1975.
- SETUBAL, J. C.; MEIDANIS, J. *Introduction to computational molecular biology*. Boston, MA, EUA: PWS Publishing Company, 320 p., 1997.

- SHMYGELSKA, A.; HOOS, H. H. An ant colony optimization algorithm for the 2D and 3D hydrophobic polar protein folding problem. *BMC Bioinformatics*, v. 6, n. 30, p. 1–22, 2005.
- SRINIVAS, N.; DEB, K. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, v. 2, n. 3, p. 221–248, 1994.
- TICONA, W. G. C.; DELBEM, A. C. B. *Algoritmos evolutivos para otimização multi-objetivo*. Nota Didática 76, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, SP, Brasil, 38 p., 2008.
- TUDELA, J. C. C.; LOPERA, J. O. Parallel protein structure prediction by multiobjective optimization. In: *PDP '09: Proceedings of the 2009 17th Euromicro International Conference on Parallel, Distributed and Network-based Processing*, Washington, DC, USA: IEEE Computer Society, 2009, p. 268–275.
- UNGER, R. The genetic algorithm approach to protein structure prediction. *Structure and Bonding*, v. 110, p. 153–175, 2004.
- UNGER, R.; MOULT, J. Finding the lowest free energy conformation of a protein is an NP-hard problem: Proof and implications. *Bulletin of Mathematical Biology*, v. 55, n. 6, p. 1183–1198, 1993a.
- UNGER, R.; MOULT, J. A genetic algorithm for 3D protein folding simulations. In: *Proceedings of International Conference on Genetic Algorithms*, 1993b, p. 581–588.
- UNGER, R.; MOULT, J. Genetic algorithms for protein folding simulations. *Journal of Molecular Biology*, v. 231, n. 1, p. 75–81, 1993c.
- VELDHUIZEN, D. A. V. *Multiobjective evolutionary algorithms: Classifications, analyses, and new innovations*. Tese de Doutorado, Department of Electrical and Computer Engineering. Graduate School of Engineering. Air Force Institute of Technology, Wright-Patterson AFB, Ohio, EUA, 250 p., 1999.
- YE, J. *A review of artificial intelligence techniques applied to protein structure prediction*. Dissertação de Mestrado, School of Computing Science, Simon Fraser University, Vancouver, Canadá, 86 p., 2007.
- YUE, K.; DILL, K. A. Forces of tertiary structural organization in globular proteins. *Proceedings of the National Academy of Sciences (USA)*, v. 92, p. 146–150, 1994.
- ZHANG, Q.; LI, H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, v. 11, p. 712–731, 2007.
- 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.
- ZITZLER, E.; LAUMANN, M.; THIELE, L. *SPEA2: Improving the strength pareto evolutionary algorithm*. Relatório Técnico 103, Computer Engineering and Networks Laboratory, Swiss Federal Institute of Technology, Zurich, Suíça, 19 p., 2001.

- ZITZLER, E.; THIELE, L. *An evolutionary algorithm for multiobjective optimization: The strength pareto approach*. Relatório Técnico 43, Computer Engineering and Communication Networks Lab, Swiss Federal Institute of Technology, Zurich, Suíça, 40 p., 1998.

Sequências de *Benchmark* Utilizadas nos Experimentos

A seguir são apresentadas as sequências de *benchmark* propostas por Unger e Moult (1993b). A Tabela A.1 traz as sequências de 27 monômeros e a Tabela A.2, as de 64 monômeros. A nomenclatura das sequências foi adotada originalmente por Patton *et al.* (1995).

Tabela A.1: Sequências de 27 monômeros.

Caso de Teste	Sequência
273d.1	HPHRHPPRHHHRHRRHHHHHHHHHHHHHHHPPH
273d.2	HPRHHHHHHHHHHHHRRHHRRHHRRHHRRHP
273d.3	PPRRHHHHHHRRHHHHRRRRHHHHHHHHHP
273d.4	PPRHHPPRRHHHRHRRHHRRHHRRHHHRP
273d.5	PPRRHHHHHRHRRHHHRHHRRHHHHHHHHH
273d.6	HHHHHHHHHHHHRRHHHRHRRHRRHRRHP
273d.7	RHHRRHRRHHHHRRHHHRHRRHRRHRRHP
273d.8	RHHHHHHHHHHHHRRHRRHHHHHHHHRRHP
273d.9	HHHHHHRRRRHHHRHRRHHHRHHRRHHH
273d.10	HHHHHRHRRHRRHRRHRRHHRRHRRHRRHH

Tabela A.2: Sequências de 64 monômeros.

Caso de Teste	Sequência
643d.1	HHRRRRHHHHRRHHHHHHRRHHHHHHHHRRHP HHHRHHRRHHHHHHHHRRHHHHRRHHRRHHRRH
643d.2	HHRRHHRRHHRRRRHHRRRRHHRRHHHHRRHH HRHHHHHHRRHHHHHHRRHHRRHHHHRRHHRRH
643d.3	RHRHHRRHHHHHHRRRRHHRRRRHHRRHHRRHP HHHRHHRRRRHHRRHHHHHHRRRRRRRRHHHH
643d.4	RHHRRHHRRHHRRHHRRHHHHHHHHHHHHRRHP RRHHRRHHHHRRHHRRHHRRHHRRHHRRHHRRHP
643d.5	RHHHRHHRRHHHHRRHHHHRRHHHHRRHHRRHP HRHHRRHHHHRRHHRRRRHHRRHHRRHHRRRRHP
643d.6	RHHRRHHRRRRHHHHHHHHRRHHRRHHHHRRHHRRH RHRHHRRHHHHHHRRHHHHHHHHHHHHRRHHRRHP
643d.7	HHHHRRHHHHRRHHRRRRHHRRHHHHHHRRHHRRH RHRHHHHHHRRHHHHHHHHHHHHRRRRHHHHRRHHHP
643d.8	HHHRRRHHRRHHRRHHRRHHHHRRHHHHRRHHRRHP HHRRHHHHHHHHRRHHRRRRHHRRRRHHRRHHRRHHHP
643d.9	RHHRRHHRRRRHHHHHHRRHHHHRRHHRRRRRRHHHH HRHRHHRRHHHHRRHHHHRRHHHHHHRRHHRRHP
643d.10	HHRRHHRRRRHHHHRRHHRRHHRRHHHHHHHHRRHH RRRRHHRRHHRRHHRRHHHHHHRRRRHHHHHHRRHHHP