CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DO PARANÁ

Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial

---

DISSERTAÇÃO

apresentada ao CEFET-PR

para obtenção do título de

MESTRE EM CIÊNCIAS

por

MARCOS PAULO SCAPIN

UM ALGORITMO GENÉTICO HÍBRIDO APLICADO À PREDIÇÃO DA ESTRUTURA DE PROTEÍNAS UTILIZANDO O MODELO HIDROFÓBICO-POLAR BIDIMENSIONAL

Banca Examinadora:

Presidente e Orientador:

PROF. DR. HEITOR SILVÉRIO LOPES	**CEFET-PR**

Examinadores:

PROF^{a}. DR^{a}. MYRIAM REGATTIERI DELGADO	**CEFET-PR**
PROF. DR. RICARDO RODRIGUES CIFERRI	**UEM**
PROF. DR. HUMBERTO M. F. MADEIRA	**PUC-PR**

Curitiba, 25 de fevereiro de 2005.

MARCOS PAULO SCAPIN

UM ALGORITMO GENÉTICO HÍBRIDO APLICADO À PREDIÇÃO DA ESTRUTURA DE PROTEÍNAS UTILIZANDO O MODELO HIDROFÓBICO-POLAR BIDIMENSIONAL

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial do Centro Federal de Educação Tecnológica do Paraná, como requisito parcial para a obtenção do título de "Mestre em Ciências" – Área de Concentração: Engenharia Biomédica.

Orientador: Prof. Dr. Heitor Silvério Lopes

Curitiba

2005

S284a Scapin, Marcos Paulo
Um algoritmo genético híbrido aplicado à predição da estrutura de proteínas utilizando o modelo hidrofóbico-polar bidimensional / Marcos Paulo Scapin. - Curitiba: [s.n.], 2005.
xviii, 132 p. : il. ; 30 cm

Orientador : Prof. Dr. Heitor Silvério Lopes
Dissertação (Mestrado) – CEFET-PR. Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial. Curitiba, 2005.
Bibliografia : p. 124-32

1. Proteínas – Estrutura – Simulação por computador. 2. Algoritmos genéticos. 3 Biologia molecular. 4. Software – Desenvolvimento. 5. Engenharia biomédica. I. Lopes, Heitor Silvério, orient. II. Centro Federal de Educação Tecnológica do Paraná. Curso de Pós-Graduação em Engenharia Elétrica e Informática Industrial. III. Título.

CDD : 574.19245
CDU : 547.96

“Welcome… to the real world.”
Morpheus (The Matrix)

“The relationship between the teaching and research
is the same as between the confession and sin:
If you have not sinned,
then you have nothing to confess!”
Anônimo

“No mundo existem 11 tipos de pessoas:
Aquelas que entendem números binários,
Aquelas que não entendem e
Aquelas que nunca vão entender.”
Anônimo

“Fly, you fools.”
Gandalf (The Lord of the Rings – The Fellowship of the Ring)

"But the power of the Ring could not be undone."
Galadriel (The Lord of the Rings – The Fellowship of the Ring)

AGRADECIMENTOS

Agradeço a Deus, por ter me dado discernimento e força durante toda a minha vida e, principalmente, nestes dois últimos anos e a Nossa Senhora, por todas as graças a mim concedidas através de sua intercessão, em especial pela conclusão de mais esta etapa de meus estudos.

A minha esposa Raquel e minha filha Bárbara, pelo apoio e dedicação oferecidos durante os momentos difíceis e trabalhosos pelos quais passei e pela compreensão pelas minhas freqüentes ausências (inclusive nos finais de semana), principalmente nos últimos meses deste mestrado.

A meus pais, Pedro e Terezinha, pelo incentivo tanto financeiro quanto moral que me motivou a enfrentar e superar mais esta fase de minha vida.

A meu orientador, Heitor, por toda a dedicação e suporte oferecidos durante a realização deste trabalho que permitiram atingir os objetivos traçados.

Aos colegas do Laboratório de Bioinformática, Denise, Guilherme, Hellen, Luciana, Maruo, Norton, Roberto, Wagner, entre outros, pelos momentos que me foram proporcionados em sua companhia.

Aos professores da banca, pelas sugestões que vieram a contribuir para o aperfeiçoamento desta dissertação.

Aos demais professores do CPGEI, pelo conhecimento transmitido durante as fases de obtenção de créditos.

A todas as outras pessoas aqui não mencionadas, não por serem menos importantes mas apenas por culpa de minha falta de memória, que tiveram participação direta e/ou indireta na realização desta dissertação.

Ao CNPq, pelo período em que me foi concedida a bolsa.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AG	– Algoritmo Genético
CGE	**–** *Charged Graph Embedding*
CHARMm	**–** *Chemistry at HARvard Molecular Mechanics*
fmGA	**–** *Fast Messy Genetic Algorithm*
GANDALF PRED	– *enhANceD genetic ALGorithm For protein structure PREDiction*
HP	**–** *Hydrophobic-Polar*
LPE	– *Lattice Polymer Embedding*
MPI	**–** *Message Passing Interface*
NMR	**–** *Nuclear Magnetic Resonance*
PDB	– *Protein Data Bank*
pfmGA	**–** *Parallel Fast Messy Genetic Algorithm*
PH	– *Perturbed Homopolymer*
SCM	– *Side Chain Model*
TSSCM	– *Tangent Spheres Side Chain Model*

RESUMO

Este trabalho propõe a utilização da técnica de computação evolucionária conhecida como algoritmos genéticos (AGs) na predição da estrutura de proteínas para o modelo 2D HP. A metodologia tem como principal proposta a utilização uma função de *fitness* melhorada, que utiliza o conceito de raio de giração. Operadores genéticos especiais foram desenvolvidos e acrescentados aos comumente usados em AG, além de novas estratégias utilizadas para auxiliar o algoritmo no processo de busca de conformações de proteínas. Estas modificações levaram ao desenvolvimento de um sistema de *software* com diversos recursos gráficos e apresentação de relatórios dos resultados, denominado GANDALF PRED. Uma certa quantidade de experimentos foi realizada com o objetivo de avaliar a influência parâmetros do AG no resultado obtido. Foram realizados dois conjuntos de testes para avaliar a metodologia proposta. O primeiro utilizou 9 seqüências de resíduos, manualmente definidas, cujos máximos de ligações são conhecidos e comprimento variando de 20 a 85 resíduos. Os resultados obtidos foram comparados com duas outras implementações encontradas na literatura. No segundo, 7 proteínas com características globulares foram escolhidas do PDB e traduzidas para o modelo HP. Seus comprimentos variam de 288 a 842 resíduos. Seus resultados foram apresentados e discutidos, já que nenhuma comparação pôde ser realizada. Para ambos os casos de teste, as conformações encontradas podem ser consideradas bons dobramentos.

ABSTRACT

This work suggests the use of an evolutionary computation technique known as genetic algorithms (GAs) for predicting protein structures in the 2D HP model. The methodology has the main proposal the use of an enhanced fitness function, which makes use of the radius of gyration concept. Special genetic operators were developed and added to those commonly used in GAs, besides new strategies to aid the algorithm in the search of protein conformations. These changes led to the development of a user-friendly software system, with several graphical resources and result reports, named GANDALF PRED. A certain amount of experiments were done with the objective of evaluating the influence of GA parameters in the result obtained. Two test cases were set to evaluate the proposed methodology. The first used 9 manually defined chains whose maximum number of hydrophobic non-local bonds is known a priori and length varying from 20 to 85 residues. The results were compared to two other implementations available in the literature. In the second, 7 proteins with globular traits were taken from PDB and translated to the HP model. Their lengths vary from 288 to 842 residues. The results were presented and discussed, since no comparison could be done. For both test cases, the conformations found can be considered good folds.

CAPÍTULO 1

INTRODUÇÃO

1.1 MOTIVAÇÕES

Uma proteína é um polímero de aminoácidos, freqüentemente chamados de resíduos, que são ligados uns aos outros através de ligações peptídicas. A função que uma determinada proteína exerce num organismo está estritamente ligada à sua estrutura tridimensional que, por sua vez, depende da seqüência de aminoácidos que a compõe.

O dobramento de proteínas é o processo pelo qual a informação linear contida na seqüência de aminoácidos de um polipeptídio dá origem à conformação tridimensional bem definida da proteína funcional. Quando as proteínas se dobram, elas adquirem uma conformação estrutural tridimensional única que é chamada de conformação nativa.

Acredita-se que a funcionalidade de uma proteína é determinada primariamente por sua conformação nativa. Desta forma, é possível realizar uma previsão de qual função uma determinada proteína desempenhará num organismo.

Porém, pode acontecer que, devido a alterações no meio (pH, temperatura, etc.), uma proteína possua estados intermediários parcialmente dobrados, inclusive estados mal-dobrados.

Quando uma grande quantidade de proteína não-dobrada se acumula, pode ocorrer agregação com outras estruturas celulares, gerando a acumulação de proteínas mal-dobradas e, como conseqüência, causar disfunção celular. Proteínas mal-dobradas não somente perdem sua função normal, mas também podem adquirir uma função totalmente diferente podendo até prejudicar a dinâmica celular.

Acredita-se que agregados destas proteínas sejam, direta ou indiretamente, a origem de condições patológicas associadas à um grande número de doenças como resultado de reações adversas no processo de dobramento das proteínas. Dentre elas, pode-se citar o Mal de Alzheimer, Mal de Parkinson, Diabetes tipo II, Doença da Vaca-Louca, vários tipos de câncer e uma quantidade de outras doenças menos conhecidas, mas não menos importantes.

Devido à relevante importância deste processo no funcionamento dos organismos e das conseqüências causadas por condições adversas decorrented do processo de dobramento,

pesquisadores de várias partes do mundo têm dedicado seus esforços na tentativa de entender como este processo realmente acontece, para que a conformação nativa de proteínas que já se conhecem possa ser determinada e, desta forma, também sua funcionalidade. Isto, juntamente com o recente seqüenciamento do genoma humano, tem gerado uma sobrecarga de informações disponíveis para a comunidade científica. Entretanto, o número de proteínas cuja conformação nativa e, consequentemente, sua funcionalidade é conhecida ainda é relativamente pequeno. Por esta razão, obter conhecimento sobre a estrutura de proteínas e, principalmente, de como elas se dobram é muito importante para que este conhecimento possa ser utilizado no desenvolvimento em laboratório de novas moléculas com funcionalidade específica desejada.

Para isto, torna-se necessária a utilização de modelos que abstraiam a formação da estrutura real de uma proteína e a apresentem em um nível de detalhes desejado. Dentre os vários modelos já propostos para este problema, o modelo Hidrofóbico-Polar (HP) é o mais simples e o mais estudado de todos, pois classifica os resíduos de aminoácidos somente em 2 tipos, além de delimitar os ângulos de rotação dos resíduos e o comprimento das ligações.

Contudo, a resolução do problema de predição da estrutura de proteínas de forma exata é muito complexa, até mesmo para os modelos mais simples, fazendo-se necessária a aplicação de métodos heurísticos na tentativa de resolver o problema. Dentre os vários métodos heurísticos existentes utilizados em implementações do problema de predição de estruturas de proteínas, aquele que vem mostrando resultados cada vez mais satisfatórios é um método chamado Algoritmos Genéticos principalmente por seu reconhecimento como uma técnica de busca eficiente e robusta.

Algoritmos Genéticos (AGs) são métodos computacionais de busca baseados nos mecanismos de evolução natural e na genética. Em AGs, uma população de possíveis soluções para o problema em questão evolui de acordo com operadores probabilísticos concebidos a partir de conceitos trazidos da biologia como, por exemplo, mutação e *crossover*, de modo que há uma tendência de que, em média, os indivíduos representem soluções cada vez melhores à medida que o processo evolutivo continua.

Portanto, espera-se que a utilização de AGs aplicados ao problema de predição da estrutura de proteínas venha a contribuir para um melhor entendimento do problema e possa impulsionar esta área de pesquisa.

1.2 OBJETIVOS

Os objetivos desta dissertação são:

- Elaborar um estudo sobre os fundamentos do processo de dobramento de proteínas e quais seus fatores relevantes;
- Analisar as técnicas já utilizadas para a resolução do problema, verificando suas vantagens e desvantagens:
  - Estudo das técnicas que tentam resolver o problema de forma exata; e
  - Estudo das técnicas heurísticas para a solução do problema;
- Implementar um Algoritmo Genético híbrido específico para o problema de predição de estruturas de proteínas utilizando o modelo HP bidimensional, desenvolvendo operadores genéticos e função de *fitness* específicos;
- Realizar experimentos com o intuito de avaliar a influência dos parâmetros do AG na predição de estruturas de proteínas;
- Realizar experimentos utilizando seqüências de proteínas tanto sinteticamente desenvolvidas quanto reais, extraídas do PDB, de modo a tentar otimizar as estruturas encontradas pelo algoritmo no modelo 2D HP;
- Avaliar os resultados obtidos pelo algoritmo implementado fazendo comparações com outras implementações e, quando não houver possibilidade de comparação, fornecer a melhor conformação obtida pelo algoritmo para determinadas proteínas reais no modelo 2D HP.

1.3 ESTRUTURA DA DISSERTAÇÃO

Esta dissertação está organizada em cinco capítulos. No Capítulo 2 apresentam-se alguns conceitos básicos de Algoritmos Genéticos (AGs). Também é apresentada uma revisão da literatura com relação aos conceitos teóricos sobre o dobramento de proteínas e como este problema tem sido abordado pelos pesquisadores. O Capítulo 3 descreve detalhadamente o desenvolvimento do algoritmo genético híbrido proposto. No Capítulo 4 relatam-se os experimentos realizados juntamente com os resultados obtidos. E, finalmente, o Capítulo 5 apresenta a discussão dos resultados, as conclusões do trabalho e as propostas de trabalhos futuros.

CAPÍTULO 2

FUNDAMENTAÇÃO TEÓRICA

2.1 ALGORITMOS GENÉTICOS

Esta seção e suas subseções foram escritas baseando-se principalmente em (GOLDBERG, 1989).

Em 1859, Charles Darwin apresentou o conceito de seleção natural, princípio segundo o qual “os indivíduos mais adaptados ao meio, apresentam maior possibilidade de sobrevivência”. Este princípio é resultado da observação de que as mais diferentes formas de vida são suscetíveis à adaptação, que ocorre através de lentas transformações genéticas, de geração em geração.

O processo de evolução ocorre através de ciclos de gerações. Cada indivíduo nasce, cresce, normalmente gera um ou mais filhos e morre. Os filhos recebem atributos genéticos dos pais. Estes atributos são responsáveis pela determinação de suas características, ou seja, seu fenótipo. Nos eucariontes, as informações genéticas estão codificadas num grande conjunto de genes, que são as unidades de transferência da hereditariedade. Cada gene pode assumir um valor particular, o alelo. Os genes formam os cromossomos, que por sua vez definem a informação genética completa de um determinado indivíduo, ou seja, o genótipo.

O conceito de algoritmo genético (AG) foi apresentado por Holland (1975) como resultado de seus estudos sobre como o fenômeno da adaptação ocorre na natureza e como este mecanismo poderia ser introduzido nos sistemas computacionais fazendo uma abstração da evolução biológica.

Os AGs são uma classe de procedimentos iterativos que simulam o processo de evolução de uma população de estruturas (indivíduos) sujeitas às forças competitivas prescritas no princípio de "sobrevivência do mais bem adaptado" de Darwin. O processo de evolução é probabilístico, porém guiado por um mecanismo de seleção baseado na adaptação de estruturas individuais.

Os AGs se diferenciam da maioria dos métodos de busca e otimização por quatro motivos:

- Trabalham em um espaço de soluções codificadas e não diretamente no espaço de busca, na maioria dos casos;
- Trabalham em um conjunto de pontos (i.e., população) e não a partir de pontos isolados;
- Não necessitam de derivadas ou outro conhecimento auxiliar, pois utilizam informações de custo ou recompensa (i.e., função objetivo);
- Usam regras de transição probabilísticas.

A execução de um AG começa com um esforço para aprender algo sobre o ambiente, ou seja, testando um número de indivíduos (população inicial) selecionados aleatoriamente do espaço de busca ou baseado num critério que capture informações sobre a estrutura do problema. Durante o processo evolutivo, cada indivíduo da população é avaliado para determinar seu valor de aptidão (*fitness*), que é a única informação utilizada pelo AG. A cada iteração do algoritmo, denominada geração, uma nova população é criada a partir de indivíduos selecionados da geração anterior mediante processos de seleção (baseando-se nos valores de *fitness* dos indivíduos) e operadores de recombinação (*crossover*) e mutação. O critério de parada de um AG pode ser definido, por exemplo, em termos do número máximo de gerações desejado, quando já houve convergência da população ou quando não há mais evolução durante uma determinada quantidade de gerações. Depois de muitas gerações, os indivíduos da população adquirem características que lhes conferem uma maior adaptabilidade ao ambiente do que os indivíduos das gerações anteriores. Quando a diferença entre as gerações é visível e mensurável, diz-se que a população evoluiu (KOZA, 1992).

O pseudocódigo a seguir (Figura 1) descreve de forma genérica um AG simples.

Algoritmo Genético

```
início
   t = 0
   inicializar P(t)
   avaliar P(t)
   enquanto não (condição de término) faça
      t = t + 1;
      selecionar P(t) a partir de P(t-1)
      recombinar e mutar P(t)
      avaliar P(t)
   fim enquanto
fim
```

Figura 1: Fluxo básico de um AG simples.

Em um AG, o termo indivíduo é aplicado a cada membro da população. Cada indivíduo possui um ou mais cromossomos (Figura 2) que contém a representação de uma possível solução para o problema sendo tratado. Os cromossomos são compostos por genes, que podem possuir diferentes valores denominados de alelos. A posição que o gene ocupa no cromossomo é denominada locus.

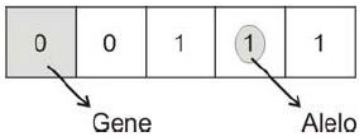


Figura 2: Exemplo de Cromossomo.

2.1.1 Codificação

De acordo com Goldberg (1989), o sucesso da execução de um AG depende de uma codificação adequada para o problema. Os cromossomos dos AGs devem, quando possível, ser codificados como uma seqüência finita em algum alfabeto finito.

Koza (1992) comenta que geralmente esta codificação é realizada sobre um alfabeto binário {0, 1}, apesar de outras codificações serem possíveis, como por exemplo, codificação em números reais. Dependendo do problema, pode existir uma grande variedade de possibilidades para codificar um indivíduo.

2.1.2 Avaliação da população

Uma das tarefas mais difíceis na modelagem dos AGs é o desenvolvimento de uma função de *fitness* apropriada, que é responsável por mensurar a qualidade de cada indivíduo da população.

Nos problemas de otimização sem restrições, o valor de *fitness* de um indivíduo pode corresponder ao valor da função objetivo. Nos problemas de otimização com restrições, a abordagem mais comum é a utilização da função de *fitness* associada a uma função de penalidade que pondera o quanto as restrições foram violadas.

2.1.3 Método de seleção

O método de seleção é usado para conduzir a evolução na direção das melhores regiões do espaço de busca. Este processo seleciona indivíduos da população para a reprodução, dando preferência aos indivíduos mais adaptados ao ambiente. Dentre os vários métodos existentes, o mais utilizado é a seleção por torneio (*tournament selection*).

A seleção por torneio não é baseada na competição entre todos os indivíduos da população, mas de um subconjunto dela. A idéia básica deste método é escolher aleatoriamente um número *t* de indivíduos da população (podendo, no entanto, representar uma porcentagem da população), selecionar o melhor indivíduo deste grupo. Em seguida, este procedimento é repetido com o intuito de selecionar mais um indivíduo, submetendo ambos aos operadores genéticos existentes, que são procedimentos que realizam alguma modificação nos indivíduos a eles submetidos. Neste caso, *t* é chamado de tamanho do torneio.

2.1.4 Operador de *crossover*

O operador de *crossover* permite a obtenção de indivíduos filhos mediante a combinação dos cromossomos dos pais. A forma mais simples é o operador de *crossover* de um ponto (Figura 3), que consiste em escolher aleatoriamente uma posição do cromossomo (ponto de *crossover*, indicado pelo pontilhado) e a partir dele trocar os genes entre os pais para formar os filhos.

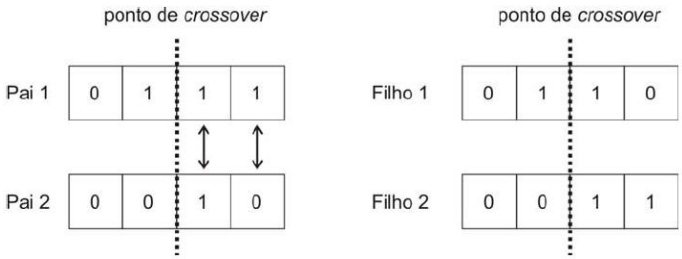


Figura 3: Exemplo de *crossover* de um ponto.

Outro operador também bastante utilizado é o *crossover* de dois pontos. A única diferença entre este e o de um ponto é a quantidade de pontos a serem escolhidos aleatoriamente. Neste caso, o segmento contido entre os pontos de *crossover* é a parte a ser trocada entre os pais, conforme mostrado na Figura 4.

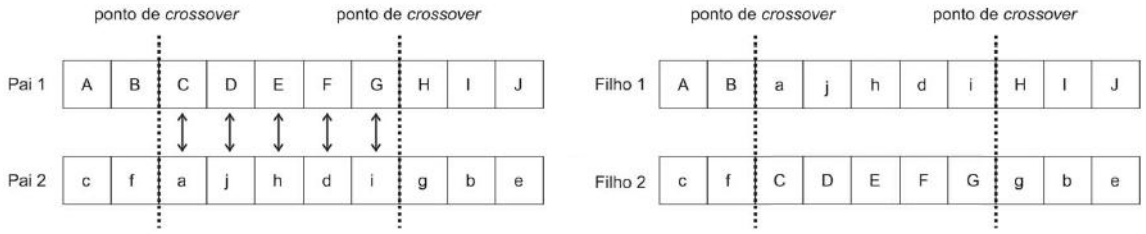


Figura 4: Exemplo de *crossover* de dois pontos.

Normalmente, o operador de *crossover* não é aplicado a todos os indivíduos selecionados para reprodução, mas somente a uma parte deles, determinada por um parâmetro chamado probabilidade de *crossover*.

2.1.5 Operador de mutação

Nos AGs, o operador de mutação executa um papel secundário, porém necessário, pois possibilita restaurar a diversidade genética eventualmente perdida durante o processo evolutivo permitindo a redução de possibilidade de convergência prematura.

A forma mais simples deste operador consiste em varrer todo o cromossomo verificando, alelo por alelo, se este deve sofrer mutação, de acordo com o parâmetro probabilidade de mutação. Caso o alelo atual deva sofrer mutação, seu valor atual é alterado para qualquer outro valor pertencente ao alfabeto escolhido. Caso contrário, o próximo alelo é testado. Um exemplo de mutação é mostrado na Figura 5.

Existem diversos outros tipos de operadores de mutação e a aplicação destes depende do tipo de codificação utilizada e do problema sendo tratado.

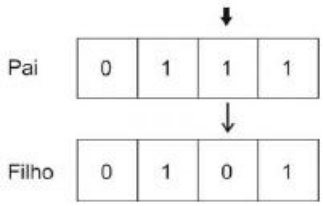


Figura 5: Exemplo de mutação.

2.1.6 Algoritmos meméticos

Algoritmos meméticos, ou algoritmos genéticos híbridos, são algoritmos evolucionários que incluem, como parte do ciclo evolutivo "padrão" de seleção-*crossover*-mutação, um estágio de busca local independente para refinar os indivíduos da população, normalmente através de operadores de busca local específicos. A aplicação de busca local traz uma considerável melhora com relação aos resultados obtidos por AGs simples quando aplicados a problemas específicos e muito difíceis, como é a predição da estrutura de proteínas (KRASNOGOR, BLACKBURNE, HIRST *et al*., 2002).

Para se obter êxito com a utilização de busca local torna-se necessário realizar um equilíbrio entre uma busca rápida (já que, normalmente, uma busca local consome bastante tempo) e o fato de manter a diversidade genética da população de modo a evitar uma convergência prematura.

Como será apresentado no capítulo 3, a implementação proposta utiliza alguns operadores de busca local com o intuito de auxiliar o processo de evolução. Levando em consideração o exposto no parágrafo anterior, uma série de experimentos serão realizados para se analisar como a utilização desta busca local afeta o comportamento do processo de evolução.

2.2 PROTEÍNAS

Uma proteína é um polímero de aminoácidos, ou seja, uma macromolécula composta de um número de subunidades semelhantes ou idênticas, chamadas monômeros, ligadas covalentemente (STANSFIELD, 1985). Os aminoácidos de proteínas são freqüentemente referidos como aminoácidos padrão, primários ou normais, para distingüi-los de outros tipos de aminoácidos, presentes em organismos vivos, mas não em proteínas (LEHNINGER, 1991). A lista dos 20 aminoácidos de proteínas com suas abreviações e fórmulas químicas pode ser consultada no ANEXO 1.

Os aminoácidos são caracterizados pela existência de um átomo de carbono central (Cα) ao qual estão ligados um átomo de hidrogênio, um grupo amina (NH_2) e um grupo carboxila (COOH). O que diferencia um aminoácido de outro são suas cadeias laterais, também chamadas de radical (R), que se encontram ligadas ao Cα (BRANDEN e TOOZE,

1999), conforme mostra a Figura 6. As cadeias laterais variam em estrutura, tamanho, carga elétrica e solubilidade em água (LEHNINGER, 1991).

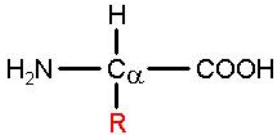


Figura 6: Estrutura de um aminoácido.

Os 20 aminoácidos diferem em sua hidrofobicidade e podem ser grosseiramente classificados em dois grupos: hidrofóbicos e hidrofílicos (LI, TANG e WINGREEN, 1997). Aminoácidos hidrofóbicos ou não-polares possuem cadeias laterais compostas de hidrocarbonos e tendem a se manter juntos na presença de água para minimizar seu contato com água. Aminoácidos hidrofílicos ou polares possuem grupos polares (com oxigênio ou nitrogênio) em suas cadeias laterais e não “se importam” em estar em contato com água (LEHNINGER, 1991; TANG, 2000).

Para formarem as proteínas, os aminoácidos devem unir-se através de ligações peptídicas (Figura 7). Na formação de uma ligação peptídica, o grupo carboxila de um dos aminoácidos perde um grupamento hidroxila (OH) enquanto o grupo amina do outro aminoácido perde um hidrogênio (H), ocorrendo a ligação entre o carbono do grupo carboxila de um aminoácido com o nitrogênio do grupo amino do outro. Os grupos OH e H liberados pelos aminoácidos em fase de ligação se unem formando uma molécula de água (H_2O). Uma molécula formada pela união de dois aminoácidos é chamada de dipeptídeo. Como uma proteína é composta por vários aminoácidos, ela pode ser considerada uma cadeia polipeptídica. As unidades de aminoácidos em um peptídeo são comumente chamadas de resíduos de aminoácidos, pois não correspondem mais aos aminoácidos originais, já que perderam um hidrogênio do grupo amina e uma porção do grupo carboxila (LEHNINGER, 1991).

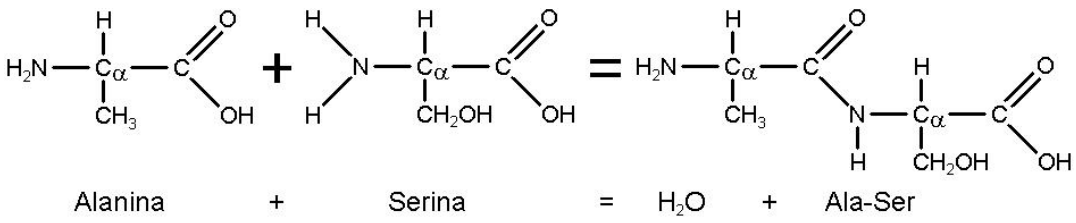


Figura 7: Exemplo de uma ligação peptídica.

Assim, o grupo amina do primeiro aminoácido de uma cadeia polipeptídica e o grupo carboxila do último aminoácido permanecem intactos, e diz-se que a orientação da cadeia tem início no grupo amina (terminal-N) e fim no grupo carboxila (terminal-C) (BRANDEN e TOOZE, 1999).

Uma cadeia peptídica é constituída por uma parte que se repete regularmente, chamada de cadeia principal, e uma parte variável, que corresponde às cadeias laterais que diferenciam os aminoácidos uns dos outros. Os átomos da cadeia principal são: um átomo de carbono (Cα) onde a cadeia lateral é ligada, um grupo NH onde o N é que fica ligado ao Cα e um grupo carboxila C=O onde o C fica ligado ao Cα (BRANDEN e TOOZE, 1999). A Figura 8a mostra um exemplo de proteína composta por 3 resíduos e a Figura 8b mostra somente sua cadeia principal.

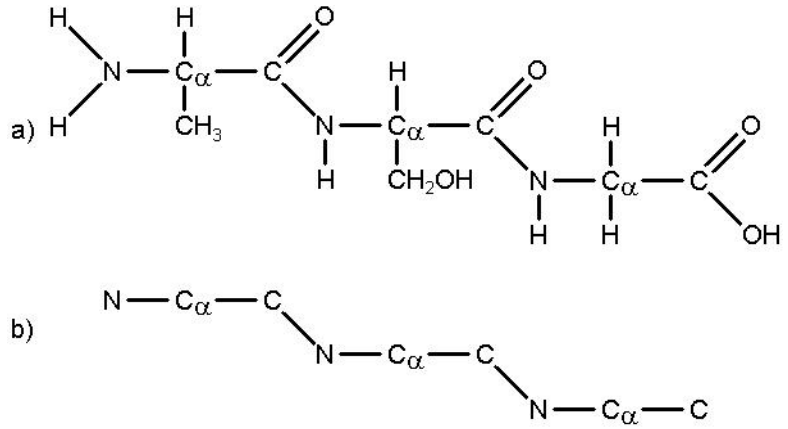


Figura 8: Exemplo de Cadeia Principal composta por 3 aminoácidos (Ala-Ser-Gly).

As proteínas exercem uma grande variedade de funções específicas nos organismos e estas funções são estritamente dependentes de suas estruturas tridimensionais (BRANDEN e TOOZE, 1999) que, por sua vez, dependem da seqüência de aminoácidos que as compõem. É, entretanto, conveniente descrever a estrutura de uma proteína de acordo com 4 diferentes níveis crescentes em complexidade (Figura 9) (LEHNINGER, NELSON e COX, 2000).

O primeiro nível, chamado de estrutura primária, corresponde à seqüência linear dos aminoácidos que constituem sua cadeia polipeptídica. Este é o nível estrutural mais simples e mais importante, pois dele deriva todo o arranjo espacial da molécula, e, conseqüentemente, sua função. A estrutura primária de duas proteínas pode diferir em número, tipos e seqüência dos aminoácidos que as compõem (LEHNINGER, NELSON e COX, 2000).

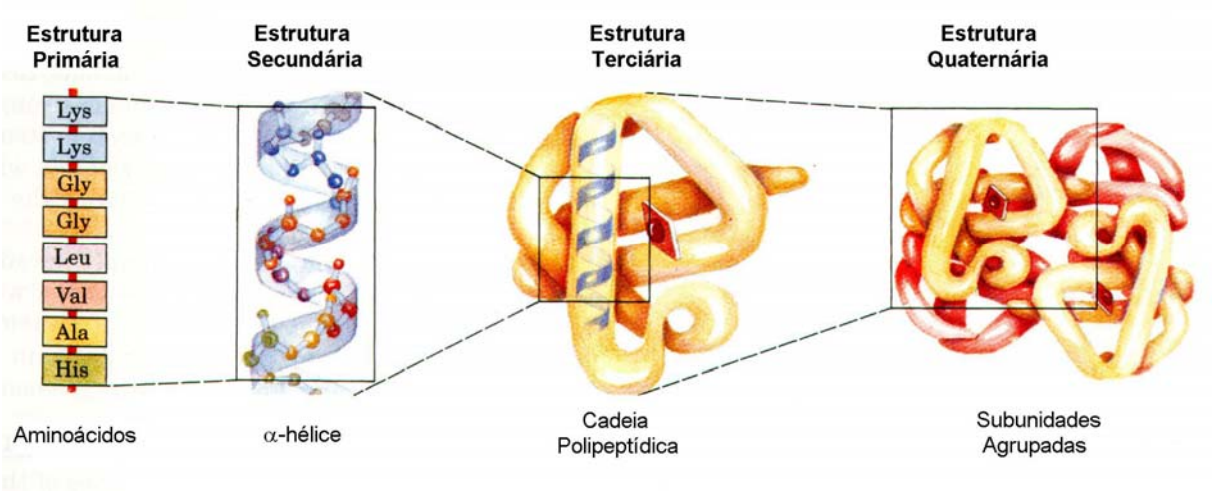


Figura 9: Exemplo das estruturas de proteínas[1].

A estrutura secundária de um segmento de uma cadeia polipeptídica é o arranjo espacial local de resíduos de aminoácidos sucessivos e próximos na cadeia polipeptídica (LEHNINGER, 1991). Este arranjo ocorre devido à possibilidade de rotação das ligações entre os carbonos dos aminoácidos e seus grupamentos amino e carboxila. O arranjo secundário de um polipeptídeo pode ocorrer de forma regular, acontecendo quando os ângulos das ligações entre carbonos e seus ligantes são iguais e se repetem ao longo de um segmento da molécula. Os dois tipos principais de arranjos secundários regulares são as α-hélices e folhas-β (Figura 10).

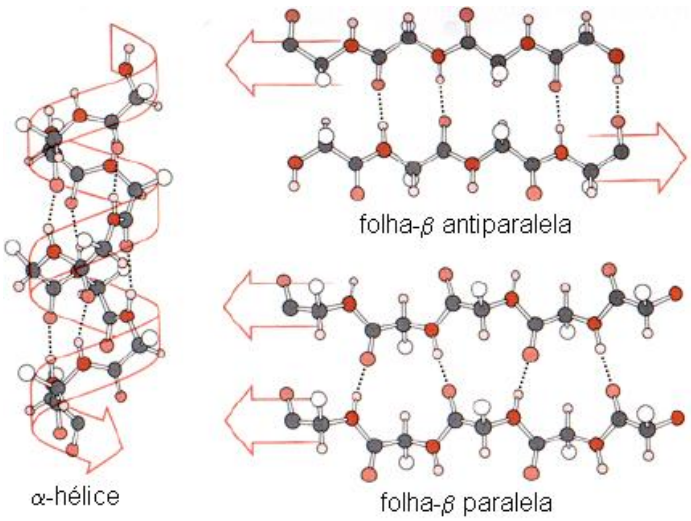


Figura 10: Exemplo de α-hélice e de folha-β.

[1] Adaptada de (LEHNINGER, NELSON e COX, 2000).

As α-hélices são as formas mais comuns de estruturas secundárias regulares. São enrolamentos do esqueleto polipeptídico em torno do eixo de uma hélice imaginária, e formadas por 3,6 resíduos de aminoácidos por volta, sendo que as cadeias laterais se projetam para fora da hélice. A estrutura é estabilizada por pontes de hidrogênio entre os grupamentos NH e CO da cadeia principal. O grupamento CO de cada aminoácido forma ponte de hidrogênio com o grupamento NH do aminoácido que está situado a quatro unidades adiante na seqüência linear, sendo que todos os grupamentos NH e CO formam pontes de hidrogênio (STANSFIELD, 1985).

As folhas-β envolvem dois ou mais segmentos polipeptídicos da mesma molécula, ou de moléculas diferentes, arranjados em paralelo (quando as cadeias adjacentes se estenderem em um mesmo sentido) ou antiparalelo (quando se estenderem em sentidos opostos). São estabilizadas por pontes de hidrogênio entre grupamentos NH e CO em fitas peptídicas diferentes, ao contrário da α-hélice cujas pontes de hidrogênio estão entre grupamentos do mesmo filamento. As cadeias laterais se alternam para cima e para baixo ao longo do esqueleto estendido.

Existem também os β-*turns* (dobras) que são responsáveis pela reversão da direção da cadeia polipeptídica. Eles são localizados, normalmente, na superfície polar, isto é, hidrofílica da proteína (UFSC, 2004).

Em média, cerca de 50% da estrutura de uma proteína globular é compreendida de α-hélices ou folhas-β. O restante da molécula assume uma estrutura secundária não repetitiva, menos regular do que as citadas.

Como um intermediário entre a estrutura secundária e terciária se encontra o que é chamada de estrutura supersecundária. Ela refere-se à combinação de elementos da estrutura secundária formando padrões que se repetem ao longo de uma mesma proteína. Por exemplo, uma folha-β separada de uma outra folha-β por uma α-hélice é encontrada em muitas proteínas. Tal padrão é chamado de unidade β-α-β. Outros padrões podem ser visualizados em (LEHNINGER, NELSON e COX, 2000).

A estrutura terciária diz respeito à forma tridimensional específica assumida pela proteína como resultado do dobramento global de toda a cadeia polipeptídica, mantendo-se a estrutura secundária apresentada. Enquanto a estrutura secundária das cadeias polipeptídicas é normalmente determinada pela interação de resíduos próximos entre si, a estrutura terciária é conferida por aspectos da interação a longa distância na seqüência de resíduos de aminoácidos (LEHNINGER, 1991). O que determina a estrutura terciária são as cadeias laterais dos

aminoácidos. Algumas cadeias são tão longas e hidrofóbicas que perturbam a estrutura secundária helicoidal, provocando dobras na proteína. As partes hidrofóbicas da proteína agrupam-se no interior da proteína dobrada, longe da água e dos íons do ambiente onde a proteína se encontra, deixando as partes hidrofílicas expostas na superfície da estrutura da proteína (UFSC, 2004).

Por fim, a estrutura quaternária descreve a forma com que as diferentes subunidades se agrupam e se ajustam para formar a estrutura total da proteína quando esta é constituída por mais de uma cadeia polipeptídica, ou seja, quando duas ou mais proteínas se acoplam para exercer sua função.

A funcionalidade de uma molécula de proteína está relacionada com as tarefas que ela realiza no organismo (processos bioquímicos de que ela participa). Acredita-se que a funcionalidade de uma proteína é determinada primariamente por sua estrutura tridimensional. Por essa razão, obter conhecimento sobre a estrutura tridimensional de proteínas e, conseqüentemente de sua funcionalidade, é importante para que este conhecimento possa ser utilizado no desenvolvimento de novas proteínas com uma funcionalidade específica desejada, como por exemplo, para curar ou controlar uma determinada doença (PEDERSEN, 2000).

2.3 DOBRAMENTO DE PROTEÍNAS

As proteínas, ao serem formadas no ribossomo, encontram-se em uma situação em que elas têm que se dobrar (*fold*) para poderem exercer sua função. O dobramento de proteínas, ou *folding*, é o processo pelo qual a informação linear contida na seqüência de aminoácidos de um polipeptídio dá origem à conformação tridimensional bem definida da proteína funcional (HARTL, 1996).

Quando em seu meio natural, ou seja, em condições fisiológicas normais, as proteínas adquirem uma conformação estrutural tridimensional única que é chamada de conformação nativa, ativa ou natural, que permite que elas realizem sua função (TANG, 2000; PEDERSEN, 2000; HENEINE, 1984). Neste estado, a proteína está com o máximo de sua organização e demonstra atividade biológica, isto é, desempenha normalmente suas funções, tais como: catálise, transporte, defesa, etc. As proteínas tendem espontaneamente para esta conformação, através de seus níveis estruturais, já citados na Seção 2.2, procurando atingir o

mais alto grau de organização, informação e eficiência de utilização de energia (HENEINE, 1984).

Como as proteínas são reais, elas devem obedecer às leis da termodinâmica que afirmam que uma biomolécula passa a maioria de seu tempo de vida em um estado de mínima energia livre (PEDERSEN, 2000). Acredita-se que a conformação nativa, por ser a estrutura mais estável de uma proteína, esteja neste estado global de mínima energia livre, o que gerou a chamada Hipótese da Termodinâmica (PEDERSEN, 2000; CHANDRU, DATTASHARMA e KUMAR, 2003).

Experimentos *in vitro* realizados por Anfinsen, Haber e White (1961) mostram que uma proteína pode ser desnaturada, ou seja, desdobrada através de alguma alteração no meio em que ela se encontra como, por exemplo, aumento da temperatura, diminuição do pH ou a redução da concentração de alguma substância desnaturante (ex. uréia) na solução. Ao retirar, lentamente, o fator de alteração da solução, fazendo o meio retornar ao seu estado inicial, a proteína tende espontaneamente a dobrar-se novamente em sua conformação nativa independentemente da conformação inicial em que se encontrava (PEDERSEN, 2000; CHANDRU, DATTASHARMA e KUMAR, 2003; TANG, 2000).

Porém, o fato de uma proteína retornar ao seu estado nativo só é válido para proteínas curtas de domínio-único (*single-domain proteins*). Domínios são regiões de uma proteína que possuem funções e estruturas tridimensionais distintas e que podem dobrar-se de forma autônoma, isto é, sem a ajuda de outras partes da proteína ou de outras proteínas (PONTIN e RUSSELL, 2002; COUNSELL, 2004a). Proteínas de domínio-único possuem seqüências de aproximadamente 100 resíduos, podendo variar de 30 a 400 resíduos (TANG, 2000; HARTL, 1996; FELDMAN e FRYDMAN, 2000).

Proteínas mais longas formam múltiplos domínios, onde cada um pode dobrar-se independentemente. Posteriormente, foi verificado que, *in vitro*, em proteínas longas de múltiplos domínios o resultado final obtido após o redobramento foi de uma grande quantidade de proteínas mal-dobradas (COUNSELL, 2004b).

Através dos experimentos de Anfinsen, verificou-se que a proteína baseia-se apenas nas informações codificadas na seqüência de aminoácidos que a compõe para que ela atinja sua conformação nativa, tanto para proteínas de domínio-único quanto para aquelas que possuem vários domínios (PEDERSEN, 2000; CHANDRU, DATTASHARMA e KUMAR, 2003; TANG, 2000).

Em experimentos *in vitro*, o redobramento espontâneo, sem assistência, para a maioria das proteínas de domínio-único, acontece da seguinte forma (DOBSON, EVANS e RADFORD, 1994; HARTL, 1996; CSERMELY, SÖTI, KALMAR *et al*., 2003):

a) Dobramento e formação da estrutura secundária: ao fazer a solução (meio) retornar a seu estado ideal, inicia-se o redobramento da cadeia polipeptídica dentro de milisegundos. Este estado parcialmente dobrado caracteriza-se por possuir sua estrutura secundária bem desenvolvida. Em geral, sua estrutura pode ser parecida com a da proteína nativa apesar da persistência das interações serem ainda bem limitadas. Poucas interações terciárias específicas estão formadas, mas já há evidências de que uma quantidade significativa de água está excluída, de acordo com a tendência de os resíduos hidrofóbicos se localizarem no interior, resultando nos chamados *molten globules*, para formar uma estrutura relativamente condensada. Com a formação de um estado compacto, o espaço conformacional que precisa ser explorado pelo restante do processo torna-se mais restrito.
b) Estabilização da estrutura secundária: dentro deste estado inicial compacto, regiões de estruturas secundárias interagem entre si tendo como resultado sua estabilização mútua. Já que estruturas secundárias nativas predominam, há uma grande probabilidade de que uma estrutura terciária mais próxima da nativa seja alcançada, sendo que ela ainda poderá ser melhorada posteriormente. A ordem em que os elementos estruturais secundários se estabilizam pode variar em detalhes de uma proteína para outra, mesmo sendo elas muito parecidas, apesar dos caminhos seguidos pelas proteínas durante o dobramento manterem suas características fundamentais.
c) Múltiplos caminhos: dado que a ordem da montagem dos elementos estruturais secundários pode ser diferente em proteínas diferentes, não há razão para se pensar que exista somente um caminho para o dobramento. Além disto, a preferência por certos tipos de estrutura secundária em relação a outros tipos é, geralmente, limitada. Mesmo que esta preferência seja aumentada pelos dobramentos hidrofóbicos, é provável que caminhos incorretos sejam explorados e “erros” precisem ser retificados. Assim, as fases mais rápidas podem representar a formação de moléculas com interações “corretas” enquanto as mais demoradas podem representar moléculas mal-dobradas em eventos anteriores.
d) Em direção à estrutura nativa: estes últimos passos do dobramento são, em geral, mais lentos que os iniciais, sendo que é somente nestes passos finais que as

cadeias laterais são compactadas. À medida que a proteína aproxima-se mais da estrutura nativa, ou seja, torna-se cada vez mais compacta, sua taxa de dobramento diminui, pois o rearranjo de suas moléculas em estados conformacionais próximos aos da conformação nativa são mais difíceis devido à diminuição dos graus de liberdade existentes, ao invés da extensão do espaço conformacional, como se pensava antigamente.

Um exemplo deste procedimento *in vitro* pode ser visualizado na Figura 11.

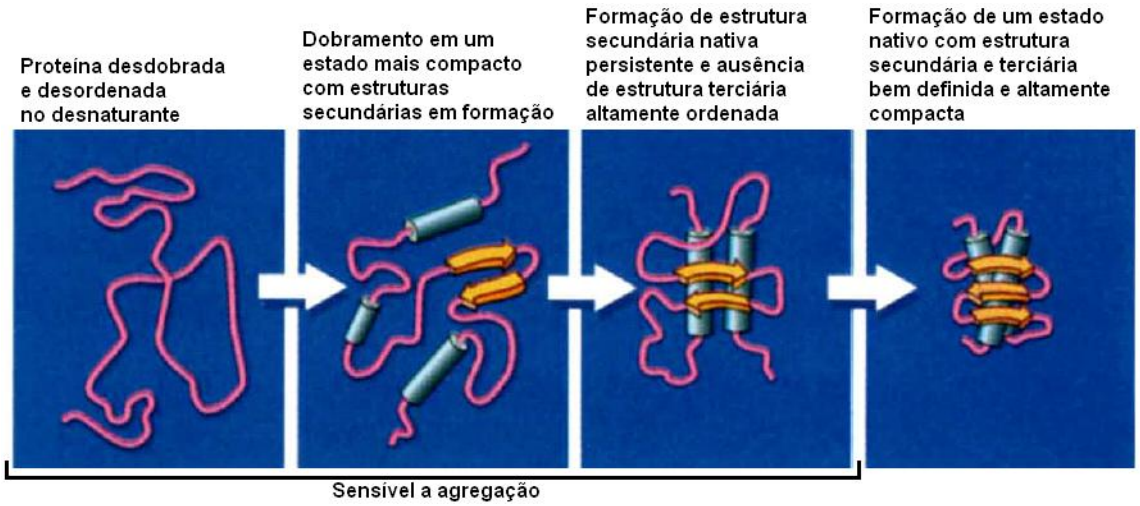


Figura 11: Exemplo de dobramento espontâneo de uma proteína de domínio-único *in vitro*[2].

Nos três primeiros estágios, ainda há um risco de a proteína não se dobrar corretamente, caso alguma partícula existente no meio venha a se agregar com ela. Este risco é eliminado após a proteína encontrar-se em seu estado nativo.

Parece improvável, mas existem dois mecanismos fundamentalmente diferentes para explicar o processo de redobramento de proteínas *in vitro* e o de dobramento *in vivo*. Certos aspectos do ambiente intracelular criam problemas para o dobramento correto de algumas proteínas recém-sintetizadas, problemas estes que são reduzidos ou eliminados do ambiente muito mais simples encontrado em experimentos *in vitro* (ELLIS e HARTL, 1999).

[2] Adaptada de (HARTL, 1996).

Um dos aspectos importantes é que no interior da célula, a parte terminal-N da cadeia polipeptídica que está sendo criada, a princípio, se dobra espontaneamente à medida que ela vai sendo sintetizada pelo ribossomo. Além disto, a taxa em que a tradução ocorre é muito menor do que a taxa de dobramento, havendo a possibilidade da cadeia sendo gerada se dobrar incorretamente antes de estar completa, se degradar porque ela não se encontra totalmente dobrada ou se agregar com cadeias próximas a ela (HARTL, 1996; ELLIS e HARTL, 1999; FELDMAN e FRYDMAN, 2000).

Entretanto, é necessário que um domínio completo já tenha sido sintetizado para que o dobramento estável possa acontecer. Desta forma, uma proteína de domínio-único iniciará seu dobramento somente após ser totalmente liberada do ribossomo, como mostra a Figura 12.

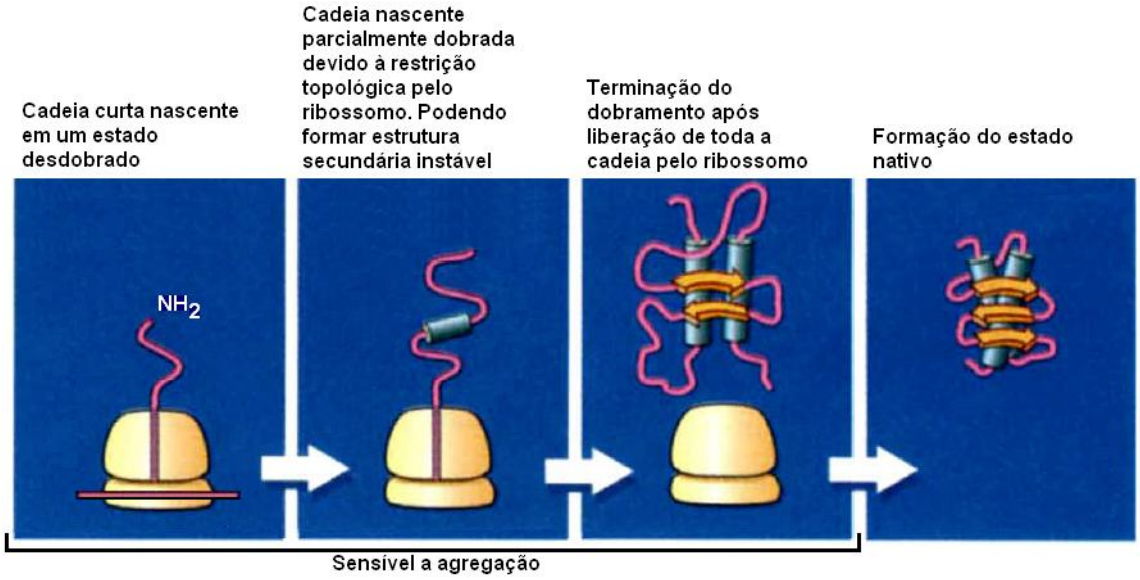


Figura 12: Exemplo de dobramento pós-tradução de uma proteína de domínio-único *in vivo*[3].

Considerando-se polipeptídios que possuem múltiplos domínios, estes terão seus domínios dobrados independentemente e à medida que eles forem sintetizados, evitando, assim, interações incorretas entre domínios durante o dobramento, conforme a Figura 13.

O dobramento de polipeptídios recém-traduzidos encontra uma limitação adicional. O interior da célula é um ambiente macromolecular onde há uma alta concentração[4] de outras macromoléculas, criando condições que aumentam bastante o risco de que cadeias recém-sintetizadas não se dobrem corretamente, mas se agreguem umas às outras formando estruturas não funcionais (ELLIS e HARTL, 1999; DOBSON e KARPLUS, 1999).

[3] Adaptada de (HARTL, 1996).
[4] Na ordem de 200–400 mg ml^{-1} de proteínas e RNA (ELLIS, 1997).

A agregação é um processo que é altamente favorecido por aumentos de concentração e temperatura, pois ela é parcialmente dirigida pela interação de regiões hidrofóbicas que ficam transitoriamente expostas na superfície de estados intermediários parcialmente dobrados. Como polipeptídios desdobrados, mesmo aqueles com alguma estrutura secundária nativa mas sem um interior hidrofóbico estável (*molten globules*), ainda expõem suas cadeias laterais hidrofóbicas, aumenta-se bastante a tendência de acontecer a agregação.

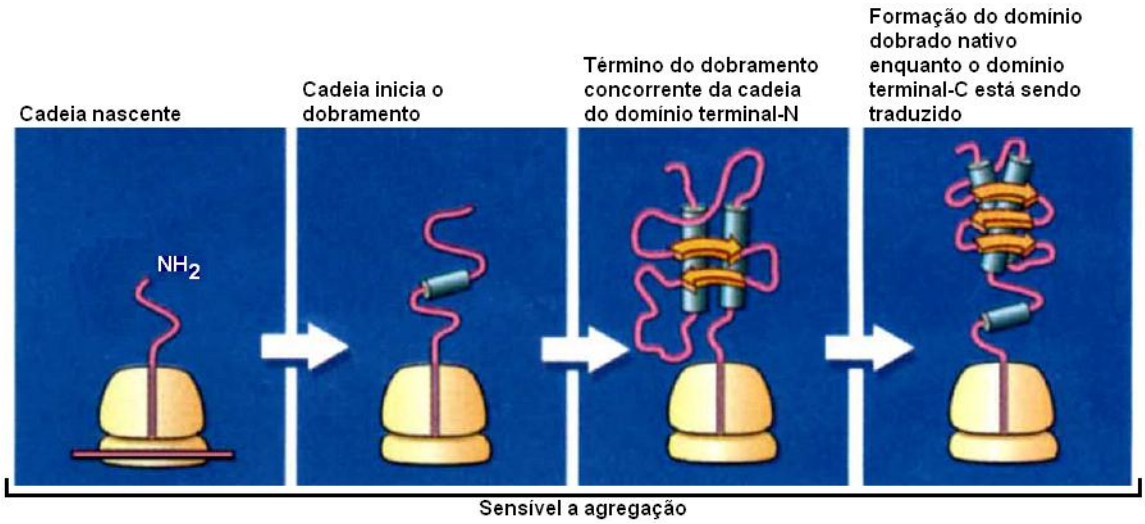


Figura 13: Exemplo de dobramento concorrente de uma proteína com dois domínios[5].

Estudos indicam que com altas concentrações celulares, as constantes de associação para macromoléculas tornam-se dez vezes maiores. Isto explica porque as células desenvolveram, através da evolução, mecanismos para assegurar que superfícies hidrofóbicas de cadeias parcialmente dobradas fiquem protegidas, retardando o início do dobramento na célula, a fim de prevenir o mal-dobramento e agregação até que uma cadeia com tamanho razoável para o dobramento correto seja sintetizada. Isto é realizado pelas chamadas chaperonas moleculares ou chaperoninas (ELLIS, 1997; ELLIS e HARTL, 1999; FELDMAN e FRYDMAN, 2000; HARTL, 1996).

As chaperoninas são grandes complexos cilíndricos que promovem o dobramento de proteínas no ambiente de sua cavidade central. Elas são proteínas que se acoplam à proteína com estrutura ainda instável (em processo de dobramento) e buscam estabilizá-la através de um processo controlado, facilitando seu dobramento correto *in vivo*, não fazendo parte da estrutura funcional final da proteína. Na realidade, elas não possuem informações que especificam o dobramento correto, mas apenas previnem que interações incorretas, tanto

[5] Adaptada de (HARTL, 1996).

internas (entre resíduos) quanto externas (com outras proteínas), ocorram enquanto a proteína não alcançou sua conformação nativa, evitando que a proteína atinja uma conformação não-nativa (HARTL, 1996).

As proteínas, ao serem traduzidas no ribossomo, têm que se dobrar mesmo que elas ainda não estejam prontas. A energia mínima livre do primeiro segmento da proteína que é liberado pelo ribossomo é, certamente, diferente da energia da proteína completa. Sendo assim, o dobramento de proteínas *in vivo* precisa ser atrasado. As chaperoninas ficam à espera da cadeia sendo traduzida e enquanto ela é liberada pelo ribossomo, as chaperoninas se acoplam a ela procurando estabilizar o polipeptídio em processo de tradução ou recém-sintetizado até que todos os segmentos necessários para o dobramento da cadeia estejam disponíveis, prevenindo o dobramento prematuro. Assim, para proteínas globulares de domínio-único, as chaperoninas retardam estas interações até que toda a proteína tenha sido sintetizada e só então é que a proteína iniciará o processo de dobramento. Para as demais proteínas, ou seja, de múltiplos domínios, este retardamento acontece para cada domínio de forma independente.

As chaperoninas também realizam o transporte de proteínas entre compartimentos subcelulares. Como os poros da mitocôndria ou do retículo endoplasmático são muito pequenos para permitir a passagem de proteínas globulares totalmente dobradas, as proteínas precisam se desdobrar para poderem atravessar a parede das organelas e, dentro delas, realizar seu redobramento a fim de voltarem a exercer sua função. Este desdobramento é auxiliado por chaperoninas que possuem a capacidade de induzir o desdobramento de complexos específicos de proteínas.

Contudo, as chaperoninas não servem somente para ajudar, mas também para destruir. Quando a quantidade de proteínas incorretamente dobradas (aquelas que perderam sua estrutura terciária e, algumas vezes, até a secundária deixando suas ligações peptídicas acessíveis) excede à capacidade dos sistemas proteolíticos intracelulares, as chaperoninas ajudam a remover estas proteínas degradadas (HARTL, 1996; CSERMELY, SÖTI, KALMAR *et al*., 2003).

Existem mecanismos (por exemplo, as chaperoninas) que buscam prevenir que uma proteína possua estados intermediários parcialmente dobrados, inclusive estados mal-dobrados, e sua tendência à agregação aumente, prevenindo, portanto, que danos sejam causados ao organismo em que elas se encontram. O mal-dobramento de uma cadeia polipeptídica pode ocorrer quando regiões, normalmente separadas na proteína, interagem durante o processo de dobramento. Estes estados não-nativos expõem resíduos hidrofóbicos e

segmentos da cadeia principal ao solvente, tendendo a se auto-associar em agregados desordenados levados por forças hidrofóbicas e pontes de hidrogênio intercadeia (DOBSON, 1999; BARRAL, BROADLEY, SCHAFFAR *et al*., 2004).

Quando altos níveis de proteínas não-dobradas ou mal-dobradas se acumulam devido a, por exemplo, calor excessivo, a capacidade de ação das chaperoninas pode exceder-se e as proteínas podem se agregar, gerando a acumulação de proteínas mal-dobradas e, como conseqüência, causar mal-funcionamento celular. Proteínas mal-dobradas não somente perdem sua função normal, mas também podem adquirir uma função totalmente diferente ou ainda prejudicar as células ao seu redor (THOMASSON, 2001; BARRAL, BROADLEY, SCHAFFAR *et al*., 2004; MOGK e BUKAU, 2004).

Assim, certa quantidade de doenças está relacionada, direta ou indiretamente, com reações adversas decorridas de problemas durante o processo de dobramento das proteínas, dentre elas o Mal de Alzheimer, Mal de Parkinson, Diabetes tipo II, doença da Vaca-Louca, vários tipos de câncer e uma quantidade de outras doenças menos conhecidas, mas não menos importantes como, por exemplo, insônia fatal de família. Estas doenças podem ser esporádicas, herdadas ou, até mesmo, infecciosas e, freqüentemente, se manifestam somente em um estágio avançado da vida. Cada doença está associada a uma proteína em particular e acredita-se que agregados destas proteínas sejam, direta ou indiretamente, a origem das condições patológicas associadas à doença em questão, sendo que em alguns casos a quantidade de material envolvido pode chegar a até alguns quilogramas de proteínas depositadas (DOBSON, 1999; THOMASSON, 2001).

2.4 O PROBLEMA DO DOBRAMENTO DE PROTEÍNAS

Devido à importância do dobramento de proteínas no funcionamento dos organismos, pesquisadores têm dedicado seus esforços ao entendimento de como esse processo realmente acontece, para que a conformação nativa de proteínas conhecidas possa ser determinada e, desta forma, também a sua funcionalidade.

Cyrus Levinthal descreveu um fato importante no dobramento de proteínas que veio a se tornar o “Paradoxo de Levinthal”: uma proteína se dobra em sua conformação nativa de energia mínima em um curto período de tempo (normalmente em um segundo ou menos, dependendo do organismo) embora o tamanho do espaço de todas as conformações possíveis de uma proteína seja muito grande, inviabilizando uma busca aleatória pela conformação

nativa, mesmo para proteínas compostas por poucos resíduos (KARPLUS, 1997; STEIPE, 1998; DOBSON e KARPLUS, 1999; DINNER, SALI, SMITH et al., 2000; CHANDRU, DATTASHARMA e KUMAR, 2003).

Determinar a seqüência de aminoácidos de uma proteína se tornou factível com a tecnologia que se tem atualmente, mas a determinação da conformação nativa (estrutura 3D) exata de uma proteína através de experimentos que usam cristalografia de raios-X ou NMR (*Nuclear Magnetic Resonance*) apresenta algumas dificuldades técnicas e financeiras, além de consumir muito tempo. Desta forma, o número de proteínas com sua estrutura tridimensional já determinada é relativamente pequeno, fazendo com que nenhuma informação estrutural esteja disponível para a vasta maioria das seqüências de proteínas conhecidas. Assim, tem havido muito interesse na tentativa de desenvolver abordagens teóricas e práticas que busquem prever a conformação nativa de proteínas utilizando-se apenas a informação contida em sua seqüência de aminoácidos (predição *ab initio*), surgindo o problema do dobramento de proteínas ou problema de predição da estrutura de proteínas (GUEX, DIEMAND e PEITSCH, 1999; LYNGSØ e PEDERSEN, 2000; CHANDRU, DATTASHARMA e KUMAR, 2003).

Nos últimos anos, o problema de predição da estrutura de proteínas tem se tornado muito diferente do problema de dobramento de proteínas. O primeiro preocupa-se com com a predição da estrutura final da proteína, enquanto o último tem seu foco nos caminhos (*pathways*) seguidos pela proteína até atingir seu estado nativo (HONIG, 1999). Neste texto ambos os termos serão utilizados indistintamente, significando, no entanto, que se trata do problema de predição da estrutura de proteínas.

Mas para se prever computacionalmente a estrutura de uma proteína é necessário ter um modelo preciso que abstraia a formação da estrutura real em um nível de detalhes desejado. Baseando-se nas leis da termodinâmica, o problema da predição de estrutura é modelado como um problema de minimização da energia livre a respeito das possíveis conformações que uma proteína pode atingir, já que esta minimização é a força mais importante para a formação da estrutura de uma proteína. A especificação de um modelo que tenha por base este princípio, chamado de modelo de energia livre, deve incluir o seguinte (PEDERSEN, 2000):

- Um modelo da proteína, ou seja, uma abstração dos átomos e das várias ligações entre eles;
- Um modelo das possíveis conformações que a proteína modelada pode assumir, ou seja, um conjunto de regras que descrevam as conformações possíveis;

- Uma função de energia computável que atribui um valor de energia livre para cada conformação possível da proteína modelada.

A solução ótima para o problema de predição de estruturas, utilizando um modelo de energia livre, é a conformação que minimiza a função de energia livre, sendo chamada de conformação nativa do modelo (PEDERSEN, 2000).

Para um modelo ser considerado relevante, ele deve apresentar algumas propriedades da formação real da estrutura. Uma delas é a equivalência visual entre as conformações nativas no modelo e as conformações nativas no mundo real. Na predição da estrutura terciária, esta propriedade corresponde à equivalência da estrutura tridimensional completa, mas para predição da estrutura secundária, a partir da estrutura tridimensional completa, as estruturas secundárias real e do modelo devem ser similares. Outra propriedade é que deve haver uma equivalência comportamental entre o processo de formação da estrutura no mundo real e o processo de formação da estrutura através do modelo (PEDERSEN, 2000).

2.5 MODELOS DE ENERGIA LIVRE

Ao utilizar um modelo de energia livre para realizar a predição da estrutura de proteínas é necessário decidir como modelar a proteína, como as possíveis conformações serão representadas e definir qual será a função de energia livre para avaliar as conformações. Isto se deve ao fato de que o nível de detalhes da estrutura a ser predita depende das escolhas que são realizadas a respeito do modelo a ser utilizado.

A Figura 14 mostra exemplos de uma mesma proteína onde o nível de detalhes é decrescente da esquerda para a direita. As duas figuras mais da esquerda mostram a estrutura de uma proteína de hemoglobina do mesmo ângulo, mas com um nível de detalhes diferente. A proteína da esquerda é representada utilizando-se todos os seus átomos, exceto os de hidrogênio, fazendo com que sua equivalência visual seja muito próxima de uma proteína real. A do meio, não leva em consideração os átomos que compõem os resíduos, apenas os resíduos como um todo. Por sua vez, a da direita, além de não considerar os átomos dos resíduos, limita que as ligações somente poderão ocorrer de acordo os ângulos determinados pela grade onde a conformação está embutida.

Figura 14: Diferentes níveis de detalhes possíveis na representação de uma proteína[6].

2.5.1 Modelos analíticos

Um modelo analítico é um modelo que possui uma descrição detalhada da estrutura da proteína, envolvendo informações sobre átomos individuais. Este modelo foi apresentado por Pedersen (2000).

Uma proteína é uma molécula que pode ser vista como uma coleção de átomos conectados uns aos outros. Para especificar a estrutura terciária de uma proteína é possível estabelecer um valor para o ângulo, o comprimento e a torção de cada ligação presente em sua estrutura, resultando em uma descrição muito complexa, pois envolve informações sobre cada átomo da proteína. Para que esta complexidade seja reduzida, alguns átomos podem ser omitidos ou agrupados em unidades maiores de forma a serem tratados como átomos individuais no modelo. Estas reduções afetam o nível de detalhes diminuindo a equivalência visual entre as conformações nativas real e do modelo.

A função de energia livre em um modelo analítico é freqüentemente especificada através de termos que indicam as contribuições feitas por átomos ligados e não-ligados. Para os átomos ligados, os termos dependem dos comprimentos, dos ângulos e das torções das ligações. Para os átomos não-ligados os termos dependem de princípios físicos (por exemplo, forças de Coulomb, de van der Walls, etc.) ou informações estatísticas inferidas de estruturas conhecidas (ex.: potenciais de força média). Entretanto, com uma descrição detalhada da estrutura da proteína e os muitos parâmetros da função de energia livre, a solução para o problema de predição de estrutura utilizando um modelo como este é computacionalmente difícil.

[6] (PEDERSEN, 2000).

Um exemplo de modelo analítico foi proposto por Ngo e Marks (1992) e apresentado por Ngo, Marks e Karplus (1994) e Chandru, Dattasharma e Kumar (2003), onde pode ser visto com maiores detalhes.

2.5.2 Modelos discretos

Um modelo muito detalhado é de pouco valor, já que o espaço de conformações possíveis é muito grande e encontrar uma conformação com energia mínima torna-se quase impossível. Este tipo de dificuldade de trabalhar com um modelo tão detalhado, como um modelo analítico, motivou os pesquisadores a desenvolverem modelos simplificados, também chamados modelos discretos, a fim de realizar a grande quantidade de simulações necessárias para obter uma descrição significativa do processo de dobramento, permitindo resultados encorajadores (DINNER, SALI, SMITH *et al*., 2000; DUAN e KOLLMAN, 2001 CHANDRU, DATTASHARMA e KUMAR, 2003). Segundo Pedersen (2000), uma maneira de reduzir a complexidade de um modelo analítico é limitar os comprimentos, ângulos e torções das ligações e utilizar somente conjuntos fixos de valores possíveis, sendo que estes valores podem ser obtidos através de estruturas reais já conhecidas, como sugerido por Pedersen e Moult (1997).

Para propósitos computacionais é desejável que as limitações dos comprimentos, ângulos e torções das ligações possíveis sejam feitas de forma que as conformações possíveis de uma proteína estejam delimitadas por uma grade, sendo que um modelo com esta propriedade é chamado de *lattice model* ou modelo treliça (PEDERSEN, 2000).

No modelo treliça há uma simplificação muito grande se comparado a um modelo analítico, pois o primeiro busca evitar detalhes dos átomos que compõem a proteína. Naquele tipo de modelo, uma proteína é modelada somente como uma seqüência de aminoácidos. Os ângulos de ligação possuem somente alguns valores discretos, definidos pela estrutura da grade. Para que uma conformação seja considerada válida, é necessário que os aminoácidos que compõem a proteína estejam limitados a uma grade, na qual cada posição da grade é ocupada por um, e somente um, aminoácido. Além disto, aminoácidos adjacentes da proteína devem ocupar posições adjacentes na grade. Por fim, a energia livre de uma conformação é especificada em função dos pares de aminoácidos não-adjacentes na proteína, mas que ocupam posições adjacentes na grade, formando as chamadas ligações não-locais (PEDERSEN, 2000; DILL, BROMBERG, YUE *et al*., 1995).

Muitos tipos diferentes de grades são possíveis, sendo que as mais utilizadas são as grades quadradas, tanto em duas quanto em três dimensões (PEDERSEN, 2000).

É perceptível que a resolução das representações das proteínas é baixa, diminuindo a equivalência visual entre as conformações reais e as do modelo, embora já tenham sido apresentados experimentos nos quais há alguma equivalência comportamental entre o processo de formação da estrutura com modelos treliça e o processo real. Também, o tamanho das cadeias utilizadas nos testes dos modelos em geral é muito pequeno, não correspondendo ao tamanho real das proteínas, embora esta limitação esteja sendo rapidamente superada (DOBSON e KARPLUS, 1999; DINNER, SALI, SMITH *et al*., 2000; PEDERSEN, 2000; DILL, BROMBERG, YUE *et al*., 1995).

Modelos treliça tornaram-se populares devido à sua simplicidade que permite que o problema de predição de estrutura seja resolvido considerando cada uma das muitas conformações possíveis de uma proteína. Isto é possível somente para proteínas pequenas, embora seja útil para o estudo de equivalência comportamental. Simulações em modelos analíticos envolvem muitos parâmetros e aproximações, tornando sua validade tão duvidosa quanto para modelos discretos (PEDERSEN, 2000; CHANDRU, DATTASHARMA e KUMAR, 2003).

Apesar de todas estas simplificações que compõem os modelos treliça, Unger e Moult (1993b), Crescenzi, Goldman, Papadimitriou *et al*. (1998) e Nayak, Sinclair e Zwick (1998), entre outros pesquisadores, provaram que o problema de predição da estrutura de proteínas para estes modelos é NP-completo, ou seja, não existe um algoritmo polinomial que consiga resolver o problema.

2.5.2.1 Modelo HP (*Hydrophobic – Polar*)

Este modelo foi introduzido por Dill (1985), sendo o modelo discreto mais conhecido e estudado de todos. Ele já foi estudado por vários pesquisadores, dentre eles Dill, Bromberg, Yue *et al*. (1995), Lyngsø e Pedersen (2000), Pedersen (2000), Tang (2000) Chandru, Dattasharma e Kumar (2003) e Scapin e Lopes (2004).

O modelo HP é a abstração mais simples possível do problema de dobramento e o modelo mais popular de grade quadrada. Ele modela o conceito de que a maior contribuição para a energia livre da conformação nativa de uma proteína deve-se a interações entre aminoácidos hidrofóbicos, que tendem a se agrupar no interior da estrutura espacial da

proteína ficando envoltos pelos aminoácidos hidrofílicos que os protegem do solvente do meio em que a proteína se encontra.

O modelo HP classifica os 20 aminoácidos em H (hidrofóbico ou não-polar) ou P (hidrofílico ou polar) baseando-se em resultados experimentais, como apontados por Li, Tang e Wingreen (1997), buscando alcançar simplicidade. Portanto, uma proteína é uma seqüência de caracteres definida sobre um alfabeto binário {H, P}. Este é um modelo treliça, pois o conjunto de conformações possíveis deve estar embutido em uma grade, neste caso, uma grade quadrada de duas ou três dimensões, sendo chamado de modelo 2D ou 3D HP, respectivamente.

Nas conformações permitidas, os aminoácidos que são adjacentes na seqüência encontram-se em posições também adjacentes na grade e nenhum ponto na grade pode ser ocupado por mais de um aminoácido.

A energia livre de uma conformação é inversamente proporcional ao número de ligações hidrofóbicas não-locais (ou ligações H–H não locais), ou seja, se uma determinada conformação possui 5 ligações H–H não-locais, diz-se que, no modelo HP, a energia desta conformação é igual a –5. Uma ligação hidrofóbica não-local é um par de aminoácidos hidrofóbicos não-adjacentes na seqüência que ocupam posições adjacentes na grade. Já que o número de pares H–H que ocorrem em posições sucessivas na proteína é fixo, a energia depende somente do número de pares H–H não-consecutivos que ficam adjacentes na grade. Desta forma, minimizar a energia livre é equivalente a maximizar o número de contatos hidrofóbicos.

A Figura 15 apresenta uma conformação que possui 6 ligações H–H não locais no modelo 2D HP. Os pontos escuros representam os resíduos hidrofóbicos e os brancos, os polares. A ligações estão representadas pelas linhas pontilhadas.

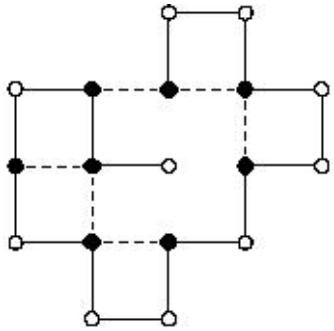


Figura 15: Exemplo de ligação H–H no modelo 2D HP.

A função de energia livre, sugerida por Li, Helling, Tang *et al*. (1996), pode ser representada pela Equação 1:

$$E = \sum_{i<j} e_{v_i v_j} \Delta(r_i - r_j) \tag{1}$$

onde:

- $\Delta(r_i - r_j) = 1$, se os resíduos r_i e r_j formam uma ligação não-local
- $\Delta(r_i - r_j) = 0$, caso contrário.

Dependendo dos tipos de contatos entre os resíduos, a energia $e_{v_i v_j}$ será e_{HH}, e_{HP} ou e_{PP}, correspondendo a contatos H–H, H–P ou P–P, respectivamente. A Figura 16 apresenta um exemplo dos tipos de contatos em um modelo 3D HP, onde os pontos escuros correspondem a Hs e os pontos claros correspondem a Ps.

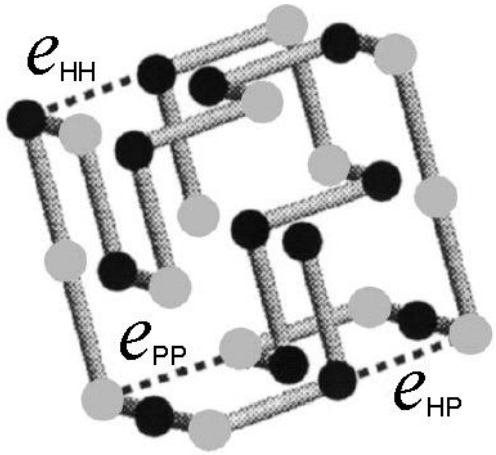


Figura 16: Exemplo do modelo 3D HP apresentando os tipos de contatos.

De acordo com Li, Helling, Tang *et al*. (1996), estes parâmetros de interação satisfazem as seguintes limitações físicas:

1. Formas compactas possuem energias menores que qualquer forma não-compacta;
2. Aminoácidos H ficam localizados no interior o máximo possível. Esta idéia é expressada na relação $e_{\mathrm{PP}} > e_{\mathrm{HP}} > e_{\mathrm{HH}}$, que diminui a energia de configurações nas quais os Hs estão escondidos da água;
3. Tipos diferentes de aminoácidos tendem a se separar, expressado pela relação $2e_{\mathrm{HP}} > e_{\mathrm{PP}} + e_{\mathrm{HH}}$.

Possíveis valores poderiam ser $e_{HH} = -2.3$, $e_{HP} = -1$ e $e_{PP} = 0$, pois satisfazem as condições 2 e 3 acima. Os resultados são insensíveis ao valor de e_{HH} desde que estas duas condições estejam satisfeitas.

Apesar da simplicidade deste modelo, o processo de dobramento no modelo possui similaridades comportamentais com o processo real e para a maioria das propriedades o modelo 2D HP tem um comportamento similar ao modelo 3D HP (DILL, BROMBERG, YUE *et al*., 1995). Portanto, sua utilização fornece uma maior contribuição no que diz respeito à realização de simulações que permitam analisar comportamentos semelhantes entre os resultados encontrados pelo modelo e as conformações reais das proteínas, não sendo tão utilizado para a realização de comparações como o modelo real, já que o grau de simplicidade é muito grande em relação à realidade. Mesmo assim, o problema foi demonstrado ser NP-completo para o modelo 2D por Crescenzi, Goldman, Papadimitriou *et al*. (1998) e para o 3D por Berger e Leighton (1998).

Algo interessante é que o modelo HP foi o primeiro modelo razoável para o qual algoritmos de aproximação para o problema de predição de estrutura de proteínas foram formulados. Isto foi feito por Hart e Istrail (1996).

Generalizações já foram consideradas para este modelo. Elas envolvem a mudança do tipo de grade, do tamanho do alfabeto e da função de energia livre. Por exemplo, Agarwala, Batzoglou, Dacík *et al*. (1997) consideram uma grade triangular além de valores proporcionais de hidrofobicidade para cada aminoácido.

2.5.2.2 Outros modelos discretos

Além do modelo HP, existem outros menos conhecidos:

- Modelo LPE (*Lattice Polymer Embedding*), formulado por Unger e Moult (1993b). Se baseia em uma grade quadrada 3D, onde o objetivo é encontrar a conformação que minimiza a energia. Unger e Moult mostram que este problema é NP-completo.
- Modelo CGE (*Charged Graph Embedding*), descrito por Ngo, Marks e Karplus (1994) e por Chandru, Dattasharma e Kumar (2003), também considera uma grade 3D. Fraenkel (1993) mostrou que este problema é NP-difícil. Uma característica

interessante do modelo CGE é que ele incorpora cobranças (*charges*) aos resíduos. Por outro lado, as conformações permitidas não são realistas.

- Modelo PH (*Perturbed Homopolymer*), apresentado por Shakhnovich e Gutin (1993) e Socci e Onuchic (1994) e revisado por Dill, Bromberg, Yue *et al*. (1995), que não leva em consideração as reações entre aminoácidos hidrofóbicos, mas favorece ligações entre aminoácidos do mesmo tipo, ou seja, H–H e P–P, desfavorecendo ligações H–P.
- Modelo HP-hélico (*helical*-HP), apresentado por Thomas e Dill (1993) e revisado por Dill, Bromberg, Yue *et al*. (1995). Este modelo, que considera apenas uma grade 2D, inclui dois tipos de interações – interações não-locais através de energia de contatos H–H e interações locais representadas por uma propensão em formar hélices, chamada de propensidade hélica.
- Modelo HP SCM (*Side Chain Model*), sugerido por Bromberg e Dill (1994), toma como referência o modelo LCM (*Linear Chain Model*) apresentado por Chan e Dill (1989a), Chan e Dill (1989b) e Lau e Dill (1989). O modelo SCM utiliza o modelo LCM para representar a cadeia principal, e acrescenta um único monômero, representando a cadeia lateral, para cada aminoácido da cadeia principal, exceto o inicial e final. Neste modelo somente as cadeias laterais são marcadas como hidrofóbicas ou hidrofílicas e não os próprios resíduos.
- Modelo HP TSSCM (*Tangent Spheres Side Chain Model*), introduzido por Hart e Istrail (1997) e utiliza tanto o modelo HP quanto o HP SCM como referência, mas não fixa os aminoácidos em uma grade. Neste modelo, o grafo que representa a proteína é transformado em um conjunto de esferas tangentes com raios iguais.

2.6 ABORDAGENS PARA O DOBRAMENTO

Como já foi dito, resolver o problema de dobramento de proteínas de forma exata é NP-difícil, mesmo para modelos muito simples como os citados nas seções anteriores. Desta forma, é natural que outros meios de investigar computacionalmente a formação da estrutura de proteínas devam ser aplicados na tentativa de obter soluções aproximadas como, por exemplo, algoritmos de aproximação, heurísticas ou outras abordagens (PEDERSEN, 2000; CHANDRU, DATTASHARMA e KUMAR, 2003).

As seções 2.6.1, 2.6.2 e 2.6.3 foram baseadas em (UNGER e MOULT, 1993d).

2.6.1 Dinâmica molecular

Esta parece ser a abordagem mais realista para a simulação do processo de dobramento real.

A idéia é simular os movimentos de cada átomo da proteína e das moléculas de água que a rodeia como uma função do tempo. É fornecida a energia térmica inicial ao sistema e os átomos podem se mover de acordo com regras da mecânica clássica. A energia de uma conformação é diferenciada para capturar as forças, acelerações e velocidades de cada átomo. Para tornar o movimento realista, é permitido que os átomos se movam somente por um curto período de tempo, em torno de 10^{-15} de segundo, antes que a energia do sistema deva ser calculada novamente, fazendo com que, mesmo em um computador de grande porte, esta simulação seja realizada por aproximadamente 10^{-9} de segundo, muitas ordens de magnitude menor que os poucos segundos que o dobramento real demora.

Há indícios de que este método de simulação produz resultados de acordo com propriedades dinâmicas observadas em proteínas (AVBELJ, MOULT, KITSON *et al.*, 1990), mas como ele só pode ser testado por um curto período de tempo, não se pode confirmar sua habilidade de convergir para a conformação nativa.

Uma revisão sobre este tipo de abordagem foi realizada por Lee, Duan e Kollman (2001).

2.6.2 Procedimentos *build-up*

A idéia básica por trás de procedimentos *build-up* é começar com um segmento curto e iterativamente dobrar o resto da cadeia.

Após pequenos segmentos terem sido analisados exaustivamente na busca por sua conformação ótima, as melhores são selecionadas como possíveis opções e os próximos resíduos são otimizados em relação a estas conformações. Novamente, as melhores possibilidades são selecionadas para servir de base para os próximos dobramentos, e assim sucessivamente até que toda a proteína seja gradualmente dobrada.

Este procedimento é muito eficiente, pois o número de opções a considerar é sempre mantido fixo e pequeno e depois que uma decisão foi tomada, o procedimento nunca volta atrás.

Porém, a principal razão para este procedimento não ter tido êxito é que nem sempre uma conformação local ótima leva a uma conformação ótima de um segmento maior. Além disto, a função de energia deve ser precisa para cada pequeno segmento e o procedimento não é robusto o suficiente para corrigir tais erros, ao contrário de outros métodos, onde a função é global e pode haver uma compensação por pequenos erros locais que venham a ocorrer.

Um exemplo de utilização deste tipo de procedimento foi apresentado por Srinivasan e Rose (2002).

2.6.3 Métodos de comparação com bases de dados

Utilizar informações de estruturas conhecidas contidas em bases de dados é uma abordagem interessante que permite melhorar o entendimento do dobramento de proteínas através de similaridades existentes. Apesar de ser uma forma diferente de predição, as informações das bases de dados podem ser utilizadas de várias formas úteis.

Uma é a modelagem homóloga, na qual a estrutura de uma proteína pode ser derivada da estrutura conhecida de outras proteínas desde que haja uma grande similaridade na seqüência de aminoácidos destas proteínas. Outras ferramentas úteis são algoritmos de detecção de *motifs* (ou seja, partes bem definidas na proteína, como por exemplo, α-hélices e folhas-β), que são capazes de detectar e utilizar similaridades locais entre famílias de proteínas para adquirir uma visão estrutural com relação a sua função. Uma outra forma é o método de construção de blocos no qual é feita uma tentativa de listar todas as possíveis conformações de pequenos segmentos da cadeia baseando-se em estruturas conhecidas.

Um exemplo de implementação pode ser visto em (XU, XU e UBERBACHER, 1998).

2.6.4 Algoritmos de aproximação

Um algoritmo de aproximação é um algoritmo que encontra soluções próximas à solução ótima com alguma garantia, ou seja, dentro de um determinado limite de erro permitido (LYNGSØ e PEDERSEN, 2000; PEDERSEN, 2000; CHANDRU, DATTASHARMA e KUMAR, 2003). Apesar dos desafios algorítmicos, a motivação para estudar algoritmos de aproximação é bem explicada por Ngo, Marks e Karplus (1994):

> "Um algoritmo de aproximação pode ter um uso prático significativo para o problema de predição da estrutura de proteínas, porque exatidão não é um requisito absoluto. Se o limite de erro garantido é suficientemente pequeno, um algoritmo de aproximação pode ser útil para gerar estruturas não precisas que, embora não dobradas corretamente, estão perto o suficiente da estrutura nativa. Caso contrário, apenas sabendo que a energia da estrutura ótima encontra-se abaixo de um certo limiar (rodando o algoritmo de aproximação) poderia ainda ser útil como parte de um esquema maior."

Os primeiros algoritmos de aproximação para o problema da predição de estruturas de proteínas foram formulados no modelo HP por Hart e Istrail (1996), sendo que algoritmos para outros modelos/grades foram desenvolvidos por Agarwala, Batzoglou, Dacík *et al*. (1997), Hart e Istrail (1997), Heun (2003), Mauri, Pavesi e Piccolboni (1999) e Newman (2002).

2.6.5 Algoritmos genéticos

Embora tenha havido algum sucesso teórico na análise de modelos discretos, mesmo que limitado, muitas pesquisas se dirigiram para análises empíricas (CHANDRU, DATTASHARMA e KUMAR, 2003).

Os esforços despendidos para resolver o problema do dobramento de proteínas quase sempre assumem que a conformação nativa corresponde ao estado global de mínima energia livre do sistema, fazendo-se necessário o desenvolvimento de técnicas eficientes de minimização de energia. A dificuldade de abordar este problema através de simulações encontra-se no fato de que o próprio problema é multi-variável e a função de energia é não-linear e multi-modal. O problema também possui uma grande quantidade de mínimos locais, fazendo com que a aplicação de métodos convencionais, como simulações por Dinâmica Molecular, não sejam eficientes e fiquem presos a mínimos locais de energia (HANSMANN e OKAMOTO, 1994; KAISER JUNIOR, LAMONT, MERKLE *et al*., 1997; OKAMOTO, 1998).

Dentre as várias abordagens já propostas para o problema do dobramento de proteínas, a mais utilizada tem sido o Algoritmo Genético (AG), principalmente por seu reconhecimento como uma técnica eficiente de busca. Utilizar um algoritmo genético não garante que o

processo encontrará a melhor solução, mas é reconhecido que à medida que a evolução avança, o AG é poderoso o suficiente para selecionar indivíduos que possuam alguma combinação de características que os faça ser mais bem adaptados ao meio (UNGER e MOULT, 1993d).

2.6.5.1 Vantagens

Segundo Unger e Moult (1993d), um algoritmo genético é uma ótima ferramenta a ser aplicada ao dobramento de proteínas de acordo com as razões descritas a seguir.

Utilizar um algoritmo genético pode possibilitar a redução da necessidade de funções de energia altamente precisas, já que as conformações são construídas gradualmente de uma maneira que subestruturas sejam avaliadas juntamente com a estrutura global, considerando-se a topologia da conformação ao invés de detalhes precisos.

Uma forte característica de busca dos algoritmos genéticos é a operação de *crossover* que permite uma varredura muito mais eficiente de regiões no espaço de conformações. Desde que alguns *motifs* locais em proteínas tenham estruturas essencialmente independentes de sua interação com o resto da molécula, eles podem evoluir com sucesso em muitos contextos durante um algoritmo genético. Assim, uma parte de estrutura que seja útil para uma solução pode ser também para outras. Mesmo se uma solução parece ser ruim ou estar errada, isto não significa que todos os seus segmentos estão errados e que todo o trabalho dedicado à criação destas soluções parciais foi desperdiçado. Desta forma, a operação de *crossover* fornece um mecanismo simples de “reciclar” soluções parciais úteis. Em termos de espaço de busca, uma combinação de características favoráveis, provavelmente, possui um valor baixo de energia livre, permitindo que através do *crossover*, com uma grande probabilidade, a busca possa ir de um mínimo diretamente para outro. Esta é uma característica importante dos algoritmos genéticos que não aparece em outros métodos de busca.

Diferentemente de outros métodos, os algoritmos genéticos não têm o compromisso de encontrar o mínimo global, mas buscam explorar as regiões mais acessíveis do espaço de conformações.

Algoritmos genéticos são convenientes para resolver problemas nos quais α-hélices e folhas-β a solução global é construída pela combinação de muitas características locais,

podendo explorar robusta e eficientemente a maneira correta de combiná-las, sendo que o problema de dobramento de proteínas é um deles.

Algoritmos genéticos não são considerados como um modelo físico para o processo de dobramento, como a dinâmica molecular, já que eles descrevem o processo em nível de população, diferentemente do processo real, que envolve uma única proteína. Assim, as muitas soluções podem ser consideradas como diferentes conformações de uma mesma molécula, e não como diferentes moléculas. Desta forma, uma operação *crossover* pode ser interpretada como uma decisão de uma única molécula, após avaliar várias possíveis conformações, em como combinar suas partes para obter uma melhor solução.

Algoritmos genéticos podem, também, ser considerados como uma variação de outros métodos aproveitando suas vantagens, enquanto evitam algumas de suas desvantagens. Por exemplo, um algoritmo genético pode ser visto como um procedimento *build-up*, já que o tamanho real das cadeias que representam as soluções é geralmente mantido fixo, mas as características requeridas de uma solução vão sendo gradualmente construídas. Enquanto aproveitam da eficiência de um procedimento *build-up*, algoritmos genéticos permitem que suas soluções sejam submetidas a avaliações globais, fazendo com que somente preferências locais que possam ter alguma influência global sejam mantidas.

Algoritmos genéticos podem ser considerados como um algoritmo de Monte Carlo paralelo, aproveitando a eficiência e robustez do método enquanto as propriedades adicionais previnem o algoritmo de permanecer em pequenas regiões de mínimos locais, explorando muitas conformações de baixa energia e escolhendo o ótimo entre elas.

2.6.5.2 Considerações sobre a implementação de um AG

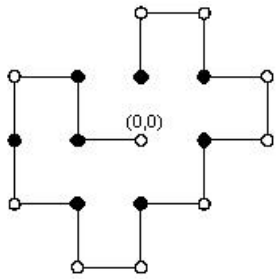
Esta seção foi escrita tomando como base os trabalhos de Piccolboni e Mauri (1998) e Krasnogor, Hart, Smith *et al*. (1999).

Na utilização de um algoritmo genético, a representação das conformações tem uma grande influência em sua dinâmica e eficiência, sendo que as três principais formas de representação propostas para o problema de dobramento são:

- Coordenadas cartesianas: a localização de cada aminoácido é especificada independentemente. Sua utilização é inadequada para algoritmos baseados em população, pois estruturas basicamente idênticas podem ter coordenadas totalmente diferentes;

- Coordenadas internas: uma conformação é especificada como uma seqüência de movimentos feitos de um aminoácido em relação ao próximo. Esta é a escolha da maioria das abordagens que usam algoritmos genéticos. Pode-se utilizar coordenadas internas absolutas ou relativas. As coordenadas absolutas são determinadas baseando-se nos eixos definidos pela grade e um exemplo seria o conjunto *{U, D, L, R, F, B}*, correspondendo a *up* (para cima), *down* (para baixo), *left* (para a esquerda), *right* (para a direita), *forward* (para frente) e *backward* (para trás). As coordenadas relativas podem ser interpretadas como a direção definida em relação ao movimento anterior cujo conjunto possível de movimentos pode ser *{U, D, L, R, F}*, sendo que o movimento *forward* corresponde à continuação no mesmo sentido do último movimento. Esta codificação exibe o problema que populações iniciais, aleatoriamente inicializadas, tendem a ter números crescentes de colisões (resíduos ocupando a mesma posição na grade) à medida que o tamanho das proteínas aumenta.
- Matriz de distância: descreve uma estrutura por meio de uma matriz de distâncias entre cada par de aminoácidos.

A Figura 17 apresenta um exemplo das formas de representação mais comuns utilizadas em AG para uma determinada conformação no modelo 2D HP para a seqüência PHHPHPHPPHPHPPHPPH. Na Figura 17a, a conformação está representada por coordenadas cartesianas. Na Figura 17b são utilizadas coordenadas internas absolutas, enquanto na Figura 17c as coordenadas são internas relativas.



a) (0,0), (-1,0), (-1,1), (-2,1), (-2,0), (-2,-1), (-1,-1), (-1,-2), (0,-2), (0,-1), (1,-1), (1,0), (2,0), (2,1), (1,1), (1,2), (0,2), (0,1)
b) E, F, E, T, T, D, T, D, F, D, F, D, F, E, F, E, T
c) E, D, E, E, F, E, D, E, E, D, E, D, E, E, D, E, E

Figura 17: Exemplo de codificação de uma conformação ao se utilizar um AG.

Segundo os resultados apresentados em (KRASNOGOR, HART, SMITH *et al*., 1999), no qual os dois tipos de coordenadas internas foram utilizados em vários tipos de grades, a codificação em coordenadas internas relativas apresentou resultados muito melhores que a codificação em coordenadas internas absolutas em grades quadradas 2D e 3D, igualando-se apenas quando uma grade triangular foi considerada.

Ao utilizar um algoritmo genético é necessário que haja algumas limitações quando da definição de uma conformação possível:

1. A conectividade da cadeia de aminoácidos, ou seja, todos os aminoácidos adjacentes na cadeia devem estar adjacentes na grade em que está definida a proteína;
2. Para uma conformação ser considerada válida é necessário que ela não possua colisões.

Talvez a maior motivação para a utilização de coordenadas internas seja a de que a primeira limitação está implícita na codificação, ao passo que com a utilização de coordenadas cartesianas, esta limitação precisa ser tratada explicitamente.

Duas abordagens básicas têm sido adotadas para lidar com a segunda limitação quando coordenadas internas são utilizadas. A primeira sugere que durante o processo de evolução somente conformações válidas, isto é, sem auto-intersecção, sejam consideradas. Este método não é muito conveniente para o problema em questão, pois o menor caminho de uma conformação válida compacta para uma outra pode ser muito mais longo se comparado com o menor caminho quando se consideram conformações inválidas intermediárias. Por exemplo, a Figura 18 apresenta uma seqüência HP que pode mudar da conformação (a) para a conformação (b) com apenas três alterações simples considerando-se conformações inválidas, sendo que se conformações intermediárias inválidas não forem permitidas, a quantidade de alterações necessárias para realizar a mudança seria muito maior.

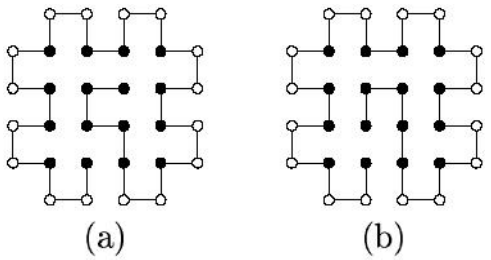


Figura 18: Duas conformações compactas “próximas” se conformações inválidas forem permitidas.

A segunda abordagem permite que conformações inválidas existam, mas utiliza penalidades para guiar o algoritmo genético a encontrar soluções válidas. Existem duas formas de penalizar conformações inválidas. Na primeira, para cada par de aminoácidos que se encontram sobre o mesmo ponto na grade, uma penalidade é acrescentada. Na segunda, uma penalidade é acrescentada para cada ponto que possua dois ou mais aminoácidos. Por exemplo, se existirem três aminoácidos sobre o mesmo ponto, a primeira forma acrescentará três penalidades, já que há três possíveis combinações de pares de aminoácidos. Considerando a segunda forma, que avalia cada ponto na grade, somente uma penalização será acrescentada. Patton, Punch III e Goodman (1995) sugerem que aminoácidos hidrofóbicos que se encontram em posições já ocupadas por outros aminoácidos não devem contribuir para a função de energia. No entanto, esta proposta pode levar a uma busca menos efetiva. Isto se deve ao fato de que quando aminoácidos hidrofóbicos são impedidos de contribuir para a função de energia por estarem sobrepostos a outros aminoácidos, o *fitness landscape* (ou *energy landscape*, que é o conjunto de todos os possíveis valores de *fitness*, ou energia, do problema sendo tratado) pode possuir grandes regiões planas tornando o problema de otimização mais difícil.

Desta forma, é recomendado o uso de uma abordagem com penalidade fixa que é adaptada baseando-se no número de aminoácidos hidrofóbicos disponíveis na seqüência de proteína, N_H. A idéia é selecionar uma penalidade suficientemente grande que assegure que qualquer conformação inválida possua energia positiva enquanto todas as conformações válidas possuam valores não-positivos de energia, fazendo com que a conformação ótima seja rigorosamente melhor do que a melhor conformação penalizada. Por exemplo, considerando uma grade quadrada, uma possível penalidade poderia ser $P = 2N_H + 2$.

Outra questão importante a ser considerada diz respeito às funções de energia. Como o modelo HP somente considera contatos H–H diretos, somente subconformações compactas contribuem para a energia destas conformações. Entretanto, é evidente que a Figura 19a está mais próxima da conformação ótima do que a Figura 19b, já que os resíduos desta conformação encontram-se mais agrupados tendo como resultado uma conformação mais compacta.

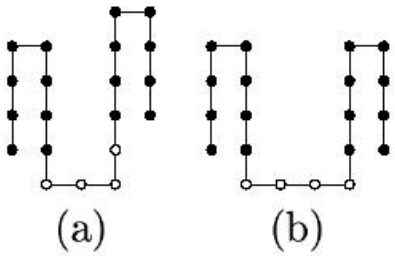


Figura 19: Duas conformações com mesma energia no modelo HP.

Este tipo de disparidade entre o valor de energia e a proximidade da conformação pode ser solucionado aumentando a função de energia para permitir que um potencial H–H dependente da distância seja considerado e, desta forma, possibilitar uma distinção a um nível mais preciso entre conformações com o mesmo número de ligações não-locais. Um exemplo poderia ser como o apresentado na Equação 2:

$$E_{H_iH_j}(d_{ij}) = \begin{cases} -1 & , d_{ij} = 1 \\ -1/\left(d_{ij}^{k} N_H\right) & , d_{ij} > 1 \end{cases} \tag{2}$$

onde:

- d_{ij} é a distância entre dois aminoácidos hidrofóbicos H_i e H_j;
- N_H é o número de aminoácidos hidrofóbicos na seqüência;
- $k = 4$ para grades quadradas e $k = 5$ para grades triangulares ou cúbicas.

2.6.5.3 Implementações

Os primeiros trabalhos nos quais AGs foram aplicados ao problema de dobramento de proteínas foram de Unger e Moult (1993a; 1993c; 1993d) que apresentam um AG que utiliza coordenadas internas absolutas em uma grade quadrada 2D e 3D para o modelo HP. Em sua implementação, uma população de conformações da cadeia polipeptídica sendo dobrada é mantida. A inicialização era não-aleatória, ou seja, todas as conformações encontram-se esticadas. O dobramento acontecia através de mutação e *crossover*, sendo que um filtro Monte Carlo era aplicado para realizar a aceitação da nova conformação de acordo com seu *fitness*. Somente conformações válidas eram consideradas, de forma que quando uma mutação e/ou *crossover* fossem aplicados, o AG realizava iterações até que uma conformação válida fosse gerada. Unger e Moult também compararam sua implementação híbrida com um algoritmo de

Monte Carlo simples e mostraram que os ganhos de desempenho que o AG proporcionou foram grandes, apesar de não ser possível sua comparação com proteínas reais.

Dandekar e Argos (1994) usaram AGs para o dobramento considerando que a proteína não estivesse embutida em uma grade, pois a proteína é representada através de coordenadas angulares internas dos átomos que pertencem à cadeia principal. Os ângulos de rotação do átomo Cα (ϕ e ψ) foram restritos a um conjunto de 7 possíveis combinações descritas por Rooman, Kocher e Wodak (1991) e mostraram-se suficientes para representar a topologia da cadeia principal em uma grande quantidade de proteínas com topologia conhecida. Cada cadeia principal foi representada por uma esfera de 1.9 Å e a função de energia consistia de pontes de hidrogênio, preferência de estrutura secundária e um termo de dispersão hidrofóbica. O método considerava a correta estrutura secundária existente para obter sucesso. Eles concluíram que interações hidrofóbicas foram mais eficientes para o dobramento do que forças locais e pontes de hidrogênio.

Schulze-Kremer e Tiedemann (1994) definiram um AG com codificação real dos ângulos ϕ e ψ e uma função de energia simplificada. O algoritmo foi aplicado a uma seqüência real, mas não obteve bons resultados. Com o intuito de encontrar a localização da cadeia lateral, o algoritmo apresentou melhores resultados.

Patton Punch III e Goodman (1995) criticam a natureza híbrida do AG apresentado por Unger e Moult e apresentam um AG padrão para o modelo 3D HP. Neste algoritmo, foi empregada a representação por coordenadas internas relativas, onde quatro mudanças repetidas de direção no mesmo sentido (à direita ou à esquerda) fazem a conformação retornar ao ponto inicial. Como este tipo de codificação tende a apresentar um grande número colisões, os autores sugeriram uma codificação expandida, na qual uma das possíveis permutações de movimentos permitidos é utilizada com o intuito de diminuir esforços para a criação de novas conformações. Um método de penalidade é utilizado para forçar conformações válidas, sendo que uma penalidade é acrescentada à função de energia se dois ou mais aminoácidos ocupam a mesma posição na grade. Qualquer aminoácido hidrofóbico que ocupe a mesma posição de um outro aminoácido não contribui para a função de energia com ligações H–H. Este AG padrão superou claramente os resultados apresentados em (UNGER e MOULT, 1993a), já que para um dado valor de energia, ele alcançou o mesmo valor em menos avaliações da função de energia e em alguns casos encontrarou melhores conformações.

Rabow e Scheraga (1996) desenvolveram um operador de combinação cartesiana e um esquema de codificação para melhorar o desempenho de AGs aplicados ao problema de

dobramento. A codificação consiste das coordenadas cartesianas dos átomos Cα da proteína. A recombinação dos genes dos pais é alcançada (1) pela rígida superposição da cadeia de um pai sobre o outro, para tornar significativa a relação de coordenadas cartesianas, e então (2) as cadeias dos filhos são formadas através da combinação linear das coordenadas de seus pais. Os filhos produzidos por este esquema de operador de combinação cartesiana possuem topologias similares e mantêm os contatos de longa distância presentes em seus pais. Eles mostram que este esquema é significativamente mais eficiente do que AGs padrão para encontrar conformações de baixa energia livre.

Dandekar e Argos (1996a; 1996b) propõem simulações do dobramento de proteínas considerando que a proteína não esteja embutida em uma grade e baseando-se em informações da seqüência de aminoácidos e conhecimento da estrutura secundária, utilizando informações tanto extraídas através de experimentos quanto analisando resultados de predições. O AG proposto utiliza a representação somente da cadeia principal através dos ângulos de rotação (ϕ e ψ). Esta abordagem é testada em proteínas cuja topologia é basicamente de α-hélices e folhas-β individualmente e, posteriormente, em proteínas que possuem topologia mista possuindo menos de 100 resíduos.

Merkle, Gaulke, Lamont *et al*. (1996) descrevem um AG híbrido que incorpora uma técnica de minimização eficiente baseada em gradiente diretamente na avaliação do *fitness*, que é baseado em um modelo geral que considera todos os átomos da proteína. O algoritmo inclui um parâmetro de freqüência de substituição que especifica a probabilidade p_r com a qual um indivíduo é substituído por sua cópia minimizada. Assim, o algoritmo pode implementar tanto evolução Baldwiniana ($p_r = 0$), Lamarckiana ($p_r = 1$) quanto Lamarckiana probabilística ($0 \leq p_r \leq 1$). Experimentos são realizados utilizando ou não o operador de minimização local. Melhores resultados são obtidos quando o operador é utilizado.

Khimasia e Coveney (1997) apresentam um AG simples para uma grade quadrada 3D que utiliza coordenadas internas absolutas que são traduzidas para coordenadas cartesianas quando do cálculo da energia livre. Para o AG proposto, duas funções de energia foram propostas (REM, *Random Energy Model*, e para o modelo HP) sendo que em ambas foram utilizadas penalidades. O estudo realizado levou a duas conclusões importantes: (1) a utilização de *crossover* de mais de um ponto é interessante, pois tende a manter os *building blocks* e (2) são necessários operadores locais para ajustar detalhes das conformações.

Kaiser, Lamont, Merkle *et al*. (1997) propõem um algoritmo de predição baseado em modelos consideram todos os átomos da proteína. Este algoritmo é expandido para incorporar conhecimento a respeito de domínios dentro do processo de busca, ou seja, é avaliada a efetividade de um AG que explora conhecimento sobre certos valores de ângulos de modo a limitar o espaço de busca. Esta abordagem é comparada com o AG híbrido que utiliza codificação binária proposto por Merkle *et al*. (1996).

Pedersen e Moult (1997) exploraram a aplicação de AGs usando uma representação atômica. Foi utilizada uma função de energia que considera forças eletrostáticas. Neste trabalho, foi utilizada uma biblioteca de ângulos, contendo os valores com maior probabilidade de aparecer, para auxiliar o processo de evolução. O método mostrou-se superior aos algoritmos de Monte Carlo analisados previamente. Para fragmentos selecionados, de até 14 resíduos, as estruturas encontradas pelo AG eram similares às correspondentes determinadas experimentalmente na maioria dos casos. Três conclusões foram obtidas: (1) o AG é um método eficaz de busca entre as conformações compactas de uma cadeia polipeptídica; (2) a função energia é geralmente capaz de selecionar conformações próximas à nativa, apesar de possuir algumas deficiências e (3) a seleção de conformações parecidas com a nativa para alguns fragmentos estabelece que nestes casos a conformação observada na estrutura da proteína completa é independente de contexto.

Em 1998, Krasnogor, Pelta, Lopez *et al*. analisaram os AGs propostos por Unger e Moult (1993c) e Patton, Punch III e Goodman (1995) e apresentaram um algoritmo que deveria combinar as piores características de ambos e, ainda assim, ser competitivo com o melhor deles. Em ambos os trabalhos anteriores, grades quadradas 3D foram utilizadas, mas neste, a grade foi reduzida a 2D devido a limitações na capacidade computacional para o modelo HP. Conformações inválidas foram permitidas, apesar de serem penalizadas. O *crossover* de dois pontos utilizado gera apenas um filho. Além de elitismo, foram utilizados quatro tipos de macromutação: após selecionar dois pontos, a parte selecionada era (a) rotacionada em 0, 90, 180 ou 270 graus; (b) refletida horizontal ou verticalmente; (c) desdobrada ou (d) alterada aleatoriamente. Os resultados obtidos sugerem que (1) *crossover* de um ponto não foi capaz de manter *building blocks* (conjunto de genes de um indivíduo que quando agrupados possuem material genético relevante) e (2) a macromutação agiu com um grande poder de busca local. Por fim, eles analisaram vários valores possíveis para os operadores, sendo que a melhor combinação indica uma probabilidade de *crossover* baixa e para mutação e macromutação alta.

Piccolboni e Mauri (1998) criticam a forma de representação utilizada por outras implementações e sugerem um AG com uma representação que se baseia em uma matriz de distâncias. Experimentos foram realizados considerando-se três operadores de *crossover* diferentes: uniforme, aritmético e em blocos. Os resultados foram comparados com os obtidos por Patton, Punch III e Goodman (1995).

Cui, Chen e Wong (1998) desenvolveram um AG que utiliza informações sobre estruturas supersecundárias de modo que estas estruturas limitem o espaço de busca. A representação utilizada considera os ângulos das cadeias principal e lateral já que os comprimentos e ângulos das ligações são mantidos fixos em seus valores ideais de acordo com a biblioteca de resíduos *Biosym*. As estruturas supersecundárias são encontradas através de uma rede neural artificial e são utilizadas na busca por estruturas de baixa energia para restringir o espaço de conformações possíveis. A função de energia possui somente dois termos: um termo que representa interações hidrofóbicas e o outro interações de van der Waals, onde o primeiro termo conduz o dobramento e o segundo é utilizado para rejeitar estruturas compactas incorretas durante o processo de dobramento.

König e Dandekar (1999a; 1999b) apresentam diferentes estratégias que buscam complementar o modelo e melhorar a eficiência de busca dos AGs para o modelo 2D HP e, posteriormente, estendido para modelo *off-lattice*. Primeiramente, são incluídas moléculas de água que envolvem a proteína sendo dobrada e as forças resultantes são consideradas na função de energia. A segunda sugestão é a chamada “busca pioneira” que melhora em 14% a eficiência do AG. A cada dez gerações, esta estratégia testa indivíduos recém-criados para verificar se eles diferem dos indivíduos já existentes na população e, caso eles não diferirem, eles são descartados. Isto possibilita explorar novas regiões do espaço de busca. A segunda estratégia chama-se “*crossover* sistemático”. Neste caso, para cada dois indivíduos selecionados para realizar o *crossover*, sendo que um deles é o melhor indivíduo, são testados todos os possíveis pontos de *crossover* e os dois melhores resultados (indivíduos) são levados para a próxima geração. Esta estratégia melhora em 50% a eficiência do AG em relação a um AG padrão, encontrando mínimos locais com melhores valores de energia e ainda é significativamente mais rápido para identificar o mínimo global.

Takahashi, Kita e Kobayashi (1999) apresentaram um AG hierárquico que combina um AG cujos genes são representados por *strings* trinárias e um AG com codificação com números reais. Na camada superior, um sub-espaço ótimo, que é definida pelos limites determinados pelos ângulos, é aplicado o AG com *strings* trinárias para realizar a busca. Na camada inferior, um valor de energia mínimo em um dado sub-espaço é procurado pelo AG

com codificação real. O método conseguiu encontrar, com sucesso, dois tipos de estruturas globais com valores de energia menores que os já conhecidos.

Sundararajan e Eils (2002) propuseram um AG que utiliza os ângulos da cadeia principal para representar as conformações. O algoritmo é inicializado com poucos aminoácidos que são evoluídos com uma população de indivíduos por um certo número de gerações. A seguir, um segundo conjunto de aminoácidos é adicionado ao primeiro acrescentando-se mais indivíduos na população com ângulos aleatoriamente definidos, aumentando, assim, o tamanho dos indivíduos. O processo continua até que todos os aminoácidos da proteína sejam incluídos e evoluídos. Os resultados apresentados não foram interessantes com relação à estrutura secundária predita, mas a terciária apresentou uma topologia bem próxima da estrutura real.

Day, Zydallis e Lamont (2002) utilizam um AG multi-objetivo, chamado *fast messy genetic algorithm* (fmGA), que utiliza a função de energia implementada pelo programa CHARMm (BROOKS, BRUCCOLERI, OLAFSON *et al.*, 1983). Ele possui a habilidade de manipular, explicitamente, *building blocks* de material genético para obter soluções boas e, potencialmente, o ótimo global. O fmGA contém três fases de operação: de inicialização, de filtragem de *building blocks* e a fase justaposicional.

Em 2003, Day, Lamont e Pachter implementaram um melhoramento do algoritmo proposto em 2002, o chamado *parallel fast messy genetic algorithm* (pfmGA). A paralelização deste algoritmo é baseada na interface MPI (*Message Passing Interface*). Ele consiste nas mesmas três fases do fmGA, todas usando comunicação síncrona. Ele opera independentemente em cada processador com comunicações somente durante as fases de inicialização e justaposicional.

Jiang, Cui, Shi et al. (2003) apresentaram um algoritmo híbrido combinando AGs e busca tabu para o modelo HP. A busca tabu é aplicada ao operador de *crossover*. Os resultados são comparados com um AG simples, com Monte Carlo evolucionário e um Monte Carlo simples e mostram que o algoritmo híbrido funciona melhor que um AG simples.

Cooper, Corne e Crabbe (2003) desenvolveram uma nova abordagem de AG com *hill-climbing* (procedimento que permite que a busca por conformações melhores seja direcionada através de algum critério previamente estabelecido). As conformações foram representadas como conjuntos de pares de números inteiros que representam os ângulos da cadeia principal. A função de energia considerava colisões entre os átomos e o colapso hidrofóbico. Os resultados obtidos foram satisfatórios além de terem sido comparados com outras implementações.

2.6.6 Outras abordagens

Várias outras abordagens já foram empregadas ao problema de dobramento de proteínas, dentre elas:

- Algoritmos meméticos (KRASNOGOR, BLACKBURNE, HIRST *et al*., 2002);
- Algoritmos multi-canônicos (HANSMANN e OKAMOTO, 1994);
- Autômatos celulares (OSTROVSKY, CROOKS, SMITH *et al*., 2001);
- Busca exaustiva (CHAN e DILL, 1991; YUE e DILL, 1993);
- Colônia de formigas (SHMYGELSKA, HERNÁNDEZ e HOOS, 2002; SHMYGELSKA e HOOS, 2003);
- Estratégias de evolução (GREENWOOD, SHIN, LEE *et al*., 1999);
- Métodos estatísticos (OSGUTHORPE, 2000; SIMON, FISER e TUSNÁDY, 2001);
- Modelos de rede (RADER e BAHAR, 2004);
- Monte Carlo e variações (O'TOOLE e PANAGIOTOPOULOS, 1992; SALI, SHAKHNOVICH e KARPLUS, 1994; HAO e SCHERAGA, 1995; NUNES, CHEN e HUTCHINSON, 1996; ORTIZ, KOLINSKI e SKOLNICK, 1998; PAPANDREOU, KANEHISA e CHOMILIER, 1998; EYRICH, STANDLEY e FRIESNER, 1999; LIANG e WONG, 2001; NAKAMURA, SASAKI e SASAI, 2001; LI, KLIMOV e THIRUMALAI, 2002; LEONHARD, PRAUSNITZ e RADKE, 2003; YESYLEVSKYY e DEMCHENKO, 2004);
- Programação por limitações (BACKOFEN, 2001);
- Redes neurais (RABOW e SCHERAGA, 1993) e fuzzy (DAUGHERITY, 1993);
- *Simulated annealing* (COVEL, 1994; HANSMANN e OKAMOTO, 1994; HIROYASU, MIKI, OGURA *et al*., 2003);
- *Simulated tempering* (IRBÄCK, 1998).

CAPÍTULO 3

METODOLOGIA

3.1 DESCRIÇÃO DO TRABALHO

Este capítulo apresenta a descrição do AG desenvolvido para realizar a predição de estruturas de proteínas utilizando o modelo 2D HP. A implementação do algoritmo descrito aqui resultou no desenvolvimento do software "GANDALF PRED" (*enhANceD genetic ALGorithm For protein structure PREDiction*), de uso restrito a aplicações acadêmicas.

3.2 CODIFICAÇÃO DOS INDIVÍDUOS

Conforme mostrado na seção 2.6.5.2, a forma como as conformações são representadas possui uma grande influência na dinâmica e eficiência de um AG. Os indivíduos representam possíveis soluções para um determinado problema e, em um AG, tais soluções são representadas indiretamente, uma vez que o conteúdo dos indivíduos representa seu genótipo. O fenótipo, ou seja, a representação da solução propriamente dita, pode ser decodificado a partir do genótipo.

Baseando-se nos resultados e discussões apresentadas por (KRASNOGOR, HART, SMITH et al., 1999) e já citados na seção 2.6.5.2, o algoritmo genético proposto utiliza coordenadas internas relativas. No espaço 2D, existem apenas três movimentos possíveis: (E)squerda, (D)ireita e (F)rente. Assim, um indivíduo é codificado sobre o alfabeto {E, D, F}. Considerando uma cadeia composta por n resíduos, os indivíduos da população possuirão $n-1$ genes, representando o número de movimentos necessários para formar uma conformação específica.

Ao utilizar AGs para tratar problemas com restrições, dependendo da representação usada, podem surgir indivíduos inválidos na população como resultado da aplicação dos operadores bem como na geração aleatória da população inicial. Nestes casos, existem três formas de lidar com esta situação: eliminar os indivíduos inválidos, corrigi-los ou permitir que eles sobrevivam na população, deixando que o processo de evolução se encarregue deles.

A primeira abordagem é simples e direta, mas pode acontecer que algum material genético importante seja perdido e não seja recuperado posteriormente. A segunda é interessante, mas computacionalmente cara, já que verificações precisarão ser feitas a cada aplicação de operadores. Na última, permite-se que indivíduos inválidos sobrevivam na população, mas o decréscimo de seu valor de fitness é proporcional às violações das restrições existentes. Esta estratégia de penalidade é útil na preservação de material genético potencialmente interessante para as gerações futuras.

Apesar das vantagens da representação escolhida (ver seção 2.6.5.2), ela permite que dois ou mais resíduos possam ocupar a mesma posição na grade. Este fato, conhecido como colisão, leva a uma conformação inválida. Entretanto, a estratégia de penalidade foi utilizada para penalizar indivíduos resultantes de dobramentos inválidos. Uma penalidade é acrescentada ao valor de *fitness* para cada ponto na grade que possua mais de um resíduo. Um outro motivo importante para permitir a existência de indivíduos que, quando decodificados, resultem em dobramentos inválidos durante a evolução é que o caminho (*path*) entre uma conformação válida compacta para uma outra conformação válida é muito mais curto se conformações inválidas forem permitidas, ao se comparar com o caminho quando somente conformações válidas sejam permitidas.

3.3 POPULAÇÃO INICIAL

De acordo com Patton, Punch III e Goodman (1995), a codificação utilizando coordenadas internas relativas exibe o problema em que a população inicial (inicializada aleatoriamente) tende a ter um grande número de colisões à medida que o tamanho da proteína aumenta. Desta forma, o AG gasta muito tempo trabalhando com conformações inválidas antes que bons resultados sejam obtidos. Para contornar esta situação, está-se propondo uma técnica diferente para a geração da população inicial, baseando-se nos conceitos extraídos de um outro tipo de algoritmo evolucionário, chamado Programação Genética (KOZA, 1992).

Esta técnica, chamada *ramped-half-and-half*, não garante que os indivíduos da população inicial estejam livres de colisões, mas com sua utilização, o número de colisões tende a diminuir além de gerar uma maior diversidade de conformações.

Para implementar esta técnica, a população é dividida em duas partes que são geradas de forma diferente. A primeira parte da população é gerada de maneira totalmente aleatória,

como é feito usualmente. A segunda é realizada como descrito a seguir. Todos os indivíduos pertencentes a esta parte da população são gerados de forma que sejam totalmente esticados, ou seja, contendo em todos os seus genes apenas o valor F. Após esta pré-inicialização, todos os indivíduos são submetidos a um número de mutações sucessivas cujo valor varia entre 3 e o número total de genes que compõem o indivíduo. Isto é realizado de forma que o número de mutações aplicadas esteja uniformemente distribuído entre todos os indivíduos sendo gerados, ou seja, deve haver o mesmo número de indivíduos com 3 mutações realizadas, com 4, e assim por diante. O mínimo de 3 mutações foi escolhido pois conformações com 0, 1 e 2 mutações possuirão poucas ligações hidrofóbicas não-locais e, consequentemente, contribuirão muito pouco para a evolução.

A proporção de cada parte em relação ao tamanho da população foi fixada internamente na implementação como sendo 20% referente à primeira parte da população e os outros 80% referente à segunda. Estes valores foram definidos empiricamente com o intuito de gerar uma população inicial com uma grande diversidade genética através de indivíduos pouco dobrados, uma condição necessária para a evolução. Foi observado através de experimentos preliminares que o processo de evolução é ajudado quando um número considerável de indivíduos possuindo poucas mutações, isto é, grandes partes não dobradas, está presente na população.

Para melhor ilustrar o funcionamento desta técnica, considere a seguinte população hipotética composta de 10 indivíduos para a seqüência “HPHPPHHPHPPHPHHPPHPH” composta por 20 resíduos. A Tabela 1 apresenta os 10 indivíduos na ordem em que foram gerados.

Tabela 1: População Inicial Hipotética (cromossomos).

Indivíduo	Cromossomo	Número Mutações
1	EEFDEDEDFEDFFFDFDEF	–
2	EDEFFFDEDFEFEEEEEDE	–
3	FFFF**E**FF**E**FFFFFFFFFF**E**	3
4	**D**FFFFF**D**FFFFFFF**D**FF**D**F	4
5	FFFF**E**F**D**F**D**FFFF**D**FFF**E**F	5
6	F**DDE**FFFFFF**E**F**DD**FFFFF	6
7	FFFFF**D**FFFF**EE**FF**DDE**F**E**	7
8	F**EEE**FF**D**F**D**F**D**F**D**FFFF**D**F	8
9	**E**F**E**F**ED**F**DDEE**FFFF**E**FFF	9
10	F**D**FF**EEE**F**D**F**D**F**DD**FF**E**F**D**	10

Pode-se perceber que os dois primeiros (20%) foram gerados de forma totalmente aleatória. Os outros (80%) apresentam uma uniformidade com relação ao número de mutações aos quais foram submetidos, sendo que cada indivíduo possui um número de mutações diferente dos outros. Caso a população fosse maior do que 20 indivíduos, haveria alguns indivíduos com a mesma quantidade de mutações. Para melhor visualizar os indivíduos, eles estão ilustrados na Figura 20, onde os pontos escuros representam os resíduos hidrofóbicos, os brancos representam os resíduos hidrofílicos e o ponto maior indica o início da seqüência. Onde houver colisões, as posições estão marcadas com um X. O X com um quadrado preenchido indica uma colisão no primeiro resíduo da seqüência.

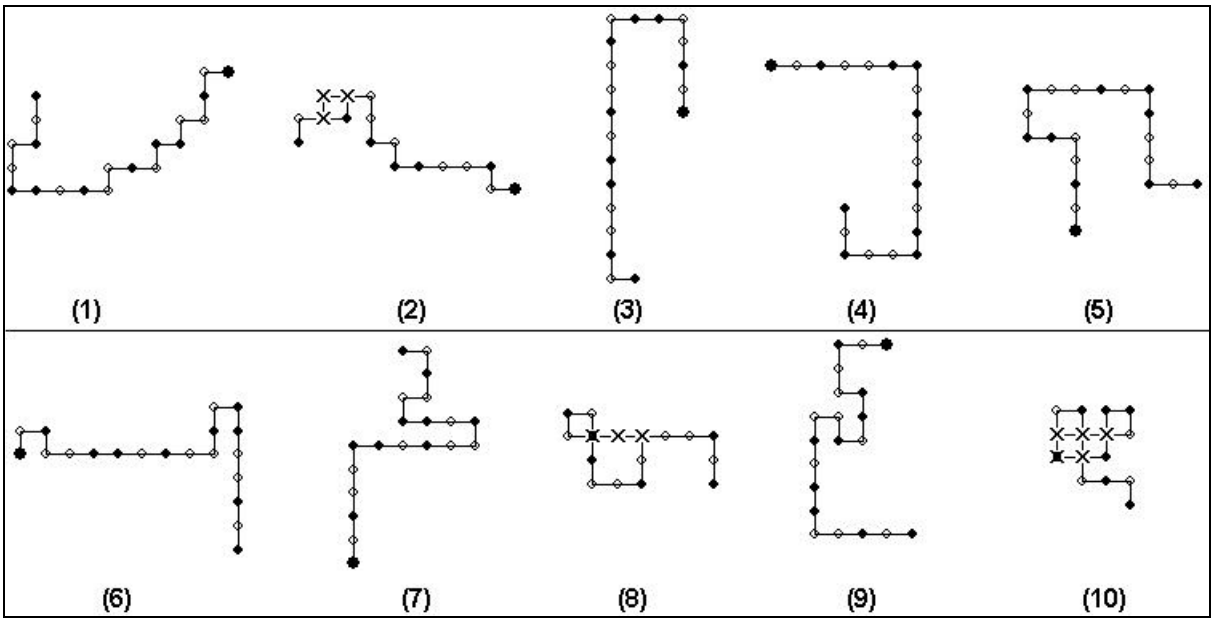


Figura 20: População inicial hipotética (indivíduos).

Para que seja possível verificar se esta técnica é realmente eficaz para gerar a população inicial, a Figura 21 mostra uma população inicial gerada de forma totalmente aleatória para a mesma proteína. Percebe-se que a maioria das conformações representa indivíduos muito dobrado e com quantidade relativa de colisões e, desta forma, tende a colaborar pouco para o processo evolutivo.

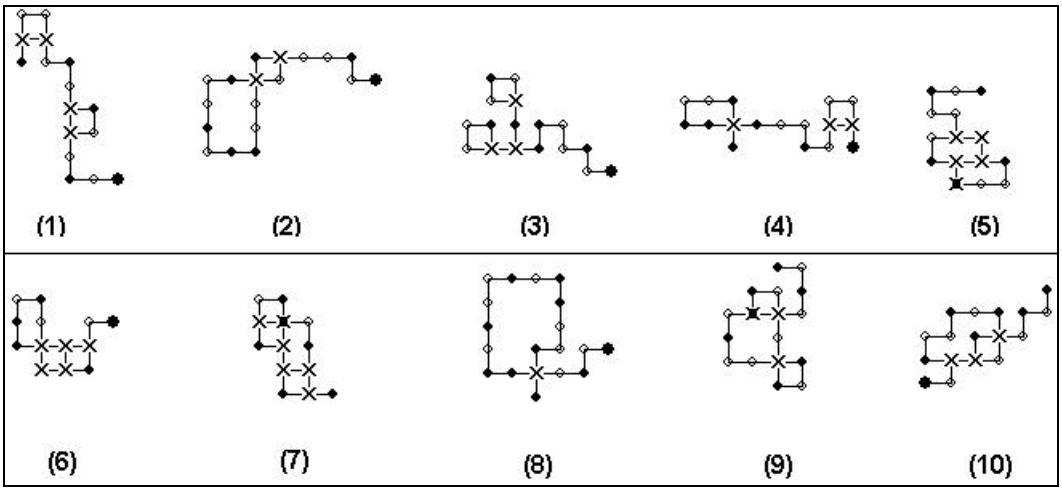


Figura 21: População inicial hipotética gerada aleatoriamente.

3.4 FUNÇÃO OBJETIVO

Para avaliar um indivíduo, é necessário decodificar sua representação genotípica (cromossomo), definida sobre o alfabeto {F, D, E}, de modo a obter as coordenadas cartesianas correspondentes a cada resíduo da proteína (representação fenotípica). Este conjunto de coordenadas descreve como os resíduos estão dispostos na grade, isto é, ele representa sua conformação bidimensional. Após esta transformação, a conformação é submetida à função objetivo para realizar sua avaliação resultando em um valor numérico que representa o grau de adaptabilidade da solução candidata dentro da população.

A função objetivo que está sendo proposta compõe-se do produto de três termos, como mostrado na Equação 3:

$$Fitness = Energia \times RadiusH \times RadiusP \tag{3}$$

onde:

- *Energia* corresponde ao termo que considera o número de ligações hidrofóbicas não-locais (*HnLB – Hydrophobic non-Local Bounds*), o número de colisões (penalidades) existentes e o peso referente a estas penalidades;
- *RadiusH* representa o raio de giração dos resíduos hidrofóbicos;
- *RadiusP* é o raio de giração dos resíduos hidrofílicos (polares).

Estes termos são explicados nas seções subseqüentes.

3.4.1 Termo *Energia*

Acredita-se que as ligações hidrofóbicas não-locais correspondem à principal força que dirige o processo de dobramento das proteínas. Conforme descrito na seção 2.5.2.1, o problema pode ser tratado tanto como de minimização da energia livre quanto de maximização de contatos H–H. O algoritmo implementado considera o problema de acordo com a última opção. Desta forma, a cada ligação hidrofóbica não-local, o valor de *HnLB* é acrescido de 1.

Relembrando que, como uma estratégia de penalidade está sendo utilizada, um termo de penalidade é subtraído diretamente de *HnLB*. Este termo é composto pelo número de pontos na grade que são ocupados por mais de um resíduo (*NC* – número de colisões) multiplicado pelo peso das penalidades (*PP*) e é apresentado na Equação 4.

$$Energia = HnLB - (NC \times PP) \tag{4}$$

O valor de *PP* é definido tendo como base o comprimento da proteína. Através de testes preliminares, percebeu-se que para uma proteína de aproximadamente 20 resíduos, uma penalidade de 2 era adequada, para uma proteína de tamanho 50, uma penalidade igual a 3 representava um valor coerente enquanto para uma de 80 resíduos, bons resultados eram obtidos com penalidade de 4. Desta forma, foi interpolada uma reta entre os pontos (20, 2), (50, 3) e (80, 4) de forma a se obter a Equação 5.

$$PP = (0,033 \times TamProt) + 1,33 \tag{5}$$

onde *TamProt* representa o número de resíduos que compõem a proteína.

3.4.2 Termo *RadiusH*

Uma questão importante a ser levada em consideração ao tentar predizer a estrutura de uma proteína está relacionada com sua *energy landscape*, ou seja, como a energia livre está distribuída ao se considerarem todas as possíveis conformações.

De acordo com Dill, Bromberg, Yue *et al.* (1995), sob condições de dobramento, não é o tamanho da *energy landscape* que importa, mas sua forma. Ela é dependente da proteína,

isto é, para cada proteína existe uma *energy landscape* diferente. Contudo, ela é multidimensional e pode possuir muitos mínimos locais.

O modelo HP original usa somente o número de ligações hidrofóbicas não-locais para avaliar os indivíduos. Entretanto, Krasnogor, Hart, Smith *et al*. (1999) argumentam que esta abordagem gera grandes regiões planas na *energy landscape*, fazendo com que uma busca local não consiga encontrar uma forma de se aproximar de um mínimo, sendo levado a uma busca aleatória. Este fato também foi confirmado nos experimentos preliminares o que motivou o desenvolvimento de uma função objetivo que tornasse a busca neste tipo de *landscape* mais fácil para o algoritmo, ao contrário da função de energia discreta descrita na Equação 4.

Deste modo, está-se propondo o uso de um novo conceito, o raio de giração (*Rg*), a ser adicionado ao cálculo da função objetivo. O uso do *Rg* na função objetivo pode ajudar o AG a escapar das regiões planas, auxiliando a busca por soluções mais compactas. Por exemplo, dadas duas conformações com o mesmo número de contatos H–H, o uso do *Rg* permite que a função objetivo diferencie uma da outra, conforme apresentado na seção 3.4.4.

O raio de giração é um conceito físico tirado da mecânica e é definido como a distância radial de um dado eixo ao qual a massa de um corpo poderia estar concentrada sem alterar a inércia rotacional do corpo em relação àquele eixo (BEER e JOHNSTON, 1980).

Trazendo o conceito de *Rg* para o problema de dobramento, ele é uma medida que indica quão compacto se encontra um conjunto de pontos (resíduos em uma grade). Conjuntos mais compactos possuem um menor valor de *Rg*.

Neste termo da função objetivo, somente os resíduos hidrofóbicos foram considerados com o intuito de medir quão compacto é o interior da conformação formada por estes resíduos. As melhores conformações são aquelas que possuem os menores valores de *Rg*. A Equação 6 mostra como *Rg* é calculado para os resíduos hidrofóbicos.

$$RgH = \sqrt{\frac{\sum_{i=1}^{NH}\left[\left(x_i - \overline{\mathrm{X}}\right)^2 + \left(y_i - \overline{\mathrm{Y}}\right)^2\right]}{NH}} \tag{6}$$

onde:

- x_i e y_i são as coordenadas cartesianas do *i*-ésimo resíduo hidrofóbico;
- $\overline{\mathrm{X}}$ e $\overline{\mathrm{Y}}$ são as médias de todos os valores x_i e y_i dos resíduos hidrofóbicos;
- NH é o número de resíduos hidrofóbicos da proteína.

Conforme já comentado, conformações compactas possuem valores baixos de *RgH*. Como o problema está sendo tratado como uma maximização, é necessário realizar a inversão deste valor de forma que conformações mais compactas possuam maiores valores. Isto é realizado conforme a Equação 7.

$$RadiusH = MaxRgH - RgH \tag{7}$$

onde $MaxRgH$ é o raio de giração calculado a partir da proteína totalmente esticada, assumindo que este seja o máximo valor que pode ser atingido.

3.4.3 Termo *RadiusP*

Este termo da função objetivo segue os mesmos conceitos apresentados na seção 3.4.2. Neste caso, somente os resíduos hidrofílicos (polares) são considerados para o cálculo do raio de giração. A intenção deste termo é fazer com que resíduos hidrofílicos se afastem do interior da estrutura dobrada.

O raio de giração dos resíduos polares é calculado da mesma maneira como é calculado para os hidrofóbicos, porém com as devidas alterações, conforme mostrado na Equação 8.

$$RgP = \sqrt{\frac{\sum_{i=1}^{NP}\left[\left(x_i - \overline{\mathrm{X}}\right)^2 + \left(y_i - \overline{\mathrm{Y}}\right)^2\right]}{NP}} \tag{8}$$

onde:

- x_i e y_i são as coordenadas cartesianas do *i*-ésimo resíduo hidrofílico;
- $\overline{\mathrm{X}}$ e $\overline{\mathrm{Y}}$ são as médias de todos os valores x_i e y_i dos resíduos hidrofílicos;
- NP é o número de resíduos polares da proteína.

Ao contrário do que acontece com os resíduos hidrofóbicos, os polares não tendem a se agrupar no interior da estrutura dobrada. Desta forma, a análise realizada baseou-se no fato de que, caso *RgP* for menor que *RgH* (os resíduos polares estiverem mais agrupados que os hidrofóbicos) para uma determinada conformação, ela sofrerá uma certa penalidade, de modo

a desfavorecer este tipo de conformação diminuindo seu valor de *fitness*, já que esta situação não é desejada.

Por outro lado, não se sabe ao certo qual é a influência do raio de giração dos resíduos hidrofílicos no dobramento de proteínas. Desta forma, quando o valor de *RgP* for maior que *RgH*, este termo não influencia em nada a função objetivo, já que *RadiusP* recebe o valor 1 e este é multiplicado pelos outros dois termos que compõem a função objetivo. A Equação 9 apresenta o cálculo realizado para a obtenção do termo *RadiusP*.

$$RadiusP = \begin{cases} 1 & \text{, se } (RgP - RgH) \geq 0 \\ \dfrac{1}{1-(RgP - RgH)} & \text{, caso contrário} \end{cases} \quad (9)$$

3.4.4 Exemplo de cálculo da função de *fitness*

Após a descrição dos termos que compõem a função de *fitness* proposta, é interessante exemplificar como seu cálculo é realizado. Para isto, serão consideradas duas conformações para a seqüência PPPHHPPHHPPPPPHHHHHHHPPHHPPPPHHPPHPP, com o mesmo número de ligações hidrofóbicas não-locais (Figura 22). Vale lembrar que os pontos escuros correspondem aos resíduos hidrofóbicos, os brancos aos resíduos polares e o ponto maior indica o início da seqüência cuja posição na grade corresponde ao ponto (0,0).

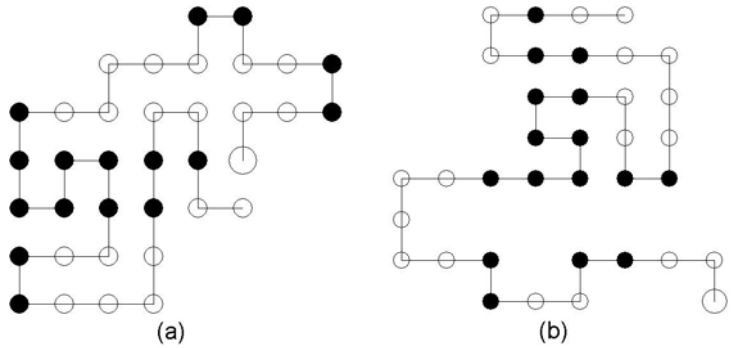


Figura 22: Duas conformações com o mesmo valor *HnLB*.

Para calcular o valor do termo *RadiusH* e *RadiusP*, torna-se necessário saber quais as coordenadas cartesianas dos resíduos que compõem as conformações. Estas coordenadas estão ilustradas na Tabela 2. As células sombreadas correspondem aos resíduos hidrofóbicos.

Tabela 2: Coordenadas cartesianas dos resíduos das conformações da Figura 22.

Ponto	Conf. A	Conf. B	Ponto	Conf. A	Conf. B
1	(0,0)	(0,0)	19	(-4,0)	(-4,4)
2	(0,1)	(0,1)	20	(-3,0)	(-4,5)
3	(1,1)	(-1,1)	21	(-3,-1)	(-3,5)
4	(2,1)	(-2,1)	22	(-3,-2)	(-2,5)
5	(2,2)	(-3,1)	23	(-4,-2)	(-2,4)
6	(1,2)	(-3,0)	24	(-5,-2)	(-2,3)
7	(0,2)	(-4,0)	25	(-5,-3)	(-1,3)
8	(0,3)	(-5,0)	26	(-4,-3)	(-1,4)
9	(-1,3)	(-5,1)	27	(-3,-3)	(-1,5)
10	(-1,2)	(-6,1)	28	(-2,-3)	(-1,6)
11	(-2,2)	(-7,1)	29	(-2,-2)	(-2,6)
12	(-3,2)	(-7,2)	30	(-2,-1)	(-3,6)
13	(-3,1)	(-7,3)	31	(-2,0)	(-4,6)
14	(-4,1)	(-6,3)	32	(-2,1)	(-5,6)
15	(-5,1)	(-5,3)	33	(-1,1)	(-5,7)
16	(-5,0)	(-4,3)	34	(-1,0)	(-4,7)
17	(-5,-1)	(-3,3)	35	(-1,-1)	(-3,7)
18	(-4,-1)	(-3,4)	36	(0,-1)	(-2,7)

As considerações feitas a seguir dizem respeito somente ao cálculo do *fitness* para a conformação A da Figura 22. A conformação B será tratada logo em seguida.

Para a realização do cálculo do termo *Energia*, é necessário se calcular o valor de *PP*, que é dependente do tamanho da seqüência ($TamProt = 36$). Assim, aplicando-se a Equação 5 a este valor tem-se o resultado na Equação 10.

$$PP = (0,033 \times 36) + 1,33 = 2,518 \tag{10}$$

De posse do valor de *PP* e sabendo que $HnLB = 6$ e $NC = 0$, é possível então obter o valor do termo *Energia* através da utilização da Equação 4, obtendo-se a Equação 11:

$$Energia = 6 - (0 \times 2,518) = 6 \tag{11}$$

O valor *MaxRgH* presente no cálculo do termo *RadiusH* pode facilmente ser calculado considerando-se a proteína totalmente esticada. Para esta proteína, $MaxRgH = 8,746$.

A aplicação da Equação 6 sobre os resíduos hidrofóbicos da conformação A, sabendo que $NH = 16$, $\overline{X} = -2,562$ e $\overline{Y} = 0,062$, resulta em $RgH = 2,838$. A Equação 12 apresenta o valor do termo *RadiusH*, calculado a partir da Equação 7.

$$RadiusH = 8,746 - 2,838 = 5,908 \tag{12}$$

Por fim, para se calcular o termo *RadiusP*, é preciso antes obter o valor de *RgP* cujo cálculo é semelhante ao *RgH*, porém somente os resíduos polares são levados em consideração. Aplicando-se os valores $NP = 20$, $\overline{X} = -1,650$ e $\overline{Y} = -0,050$ na Equação 8, obtém-se $RgP = 2,424$.

Submetendo-se os valores de *RgH* e *RgP* à condição da Equação 9, constata-se que a diferença $RgP - RgH$ é negativa. Sendo assim, o valor do termo *RadiusP* é mostrado na Equação 13.

$$RadiusP = \frac{1}{1-(2,424-2,838)} = \frac{1}{1-(-0,414)} = \frac{1}{1,414} = 0,707 \tag{13}$$

Desta forma, o valor do *fitness* da conformação A da Figura 22 corresponde ao apresentado da Equação 14.

$$Fitness = 6 \times 5,908 \times 0,707 \approx 25,062 \tag{14}$$

Com relação à conformação B, verifica-se que o valor do termo *Energia* é o mesmo do da conformação A (Equação 11).

Sabendo que $NH = 16$, $\overline{X} = -3,437$ e $\overline{Y} = 3,437$, para os resíduos hidrofóbicos desta conformação, tem-se $RgH = 2,262$. Portanto, o valor do termo *RadiusH* para esta conformação é mostrado na Equação 15.

$$RadiusH = 8,746 - 2,262 = 6,484 \tag{15}$$

Com relação aos resíduos polares, tem-se $NP = 20$, $\overline{X} = -3,250$ e $\overline{Y} = 3,450$. Desta forma, $RgP = 3,440$. Ao submeter os valores de *RgH* e *RgP* à condição da Equação 9, verifica-se que a primeira condição é satisfeita, ou seja, $RadiusP = 1$. Sendo assim, o valor de *fitness* para a conformação B está expressa na Equação 16.

$$Fitness = 6 \times 6,484 \times 1 \approx 38,904 \tag{16}$$

Percebe-se, no entanto, que a função de *fitness* é capaz de diferenciar duas conformações com o mesmo número de ligações, favorecendo aquela que possui seus resíduos hidrofóbicos agrupados mais no interior da conformação.

De modo a compreender melhor a influência de cada fator da função de *fitness* no processo de avaliação dos indivíduos, a Figura 23 ilustra quatro conformações para a mesma proteína descrita no início desta seção e a Tabela 3 apresenta os valores de cada um dos parâmetros utilizados no cálculo da função de *fitness*.

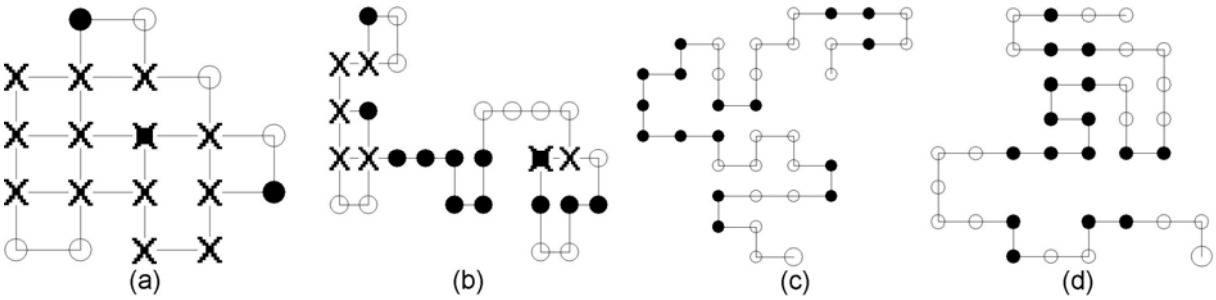


Figura 23: Diferentes conformações para uma mesma proteína.

Observando-se a Tabela 3 percebe-se que, apesar do valor de *RgH* para a conformação A ser menor que o da B, a primeira possui quase o dobro de colisões. Neste caso, o termo *Energia* (mais propriamente o valor *NC*) possui uma grande influência no valor do *fitness* final do indivíduo. Desta forma, quando uma conformação possui muitas colisões ($Energia < 0$), é interessante que esta conformação possua valores de *RadiusH* menores, de modo que o resultado da multiplicação dos termos resulte em valores maiores. No entanto, quando as conformações não possuem colisões (C e D), ambos os termos, *Energia* e *RadiusH*, contribuem significativamente para o valor do *fitness* final. Além disto, o termo *RadiusP*, quando não possui valor 1, funciona como atenuante do resultado.

Tabela 3: Valores dos termos do *fitness* para as conformações da Figura 23.

Conformação / **Termos**	**A**	**B**	**C**	**D**
HnLB	0	2	2	6
NC	13	7	0	0
Energia	–32,734	–15,626	2	6
RgH	1,476	3,040	2,941	2,262
RadiusH	7,270	5,706	5,805	6,484
RgP	1,715	3,714	2,939	3,440
RadiusP	1	1	0,998	1
Fitness	–237,976	–89,162	11,586	38,904

3.5 OPERADORES GENÉTICOS BÁSICOS

Os operadores genéticos são usados para modificar os indivíduos da população tendo como objetivo criar novos indivíduos. Sendo assim, torna-se necessária a utilização de um método que selecione indivíduos da população atual de acordo com algum critério previamente estabelecido. O método de seleção não é um operador genético, mas um procedimento que deve ser executado antes de sua aplicação.

A implementação proposta utiliza o método de seleção por torneio estocástico, conforme descrito na seção 2.1.3.

3.5.1 *Crossover*

Este é o primeiro operador genético ao qual são submetidos os indivíduos selecionados pelo método de seleção. De acordo com (UNGER e MOULT, 1993d), este operador é uma das principais características de busca dos AGs, já que parte de uma estrutura pode ser tão útil para outras conformações quanto foi para aquela que a possui.

Após serem selecionados dois indivíduos, precisa-se verificar se eles serão realmente submetidos a este operador, de acordo com a probabilidade de *crossover*. Caso devam ser submetidos, o algoritmo escolhe, aleatoriamente (ou seja, com 50% de probabilidade), qual dos dois operadores de *crossover* será aplicado, de 1 ou 2 pontos.

Para a aplicação do *crossover* de 1 ponto, o algoritmo escolhe uma posição do cromossomo, definida como ponto de *crossover*, dividindo o indivíduo em duas partes. Assim, todos os genes, a partir desta posição, serão trocados entre os pais, de forma que dois novos indivíduos (filhos) sejam gerados. O primeiro filho será composto da parte inicial do primeiro pai e da parte final do segundo. O segundo será composto da parte inicial do segundo pai e da parte final do primeiro, conforme ilustrado na Figura 24.

Figura 24: Aplicação de *crossover* de 1 ponto.

Caso o algoritmo selecione o *crossover* de dois pontos, ele deverá escolher 2 pontos de cruzamento, dividindo os indivíduos em 3 partes. Neste caso, os genes existentes entre as duas posições selecionadas serão trocados entre os pais de modo a gerar os dois novos filhos. A Figura 25 ilustra o procedimento descrito.

Figura 25: Aplicação de *crossover* de 2 pontos.

3.5.2 Mutação

Mutação é um outro operador comumente usado em algoritmos evolucionários. Após a aplicação, ou não, do operador de *crossover*, cada um dos indivíduos recém-gerados (filhos) são submetidos ao operador de mutação. Como dois tipos de mutação foram implementados, o algoritmo inicialmente seleciona com alguma probabilidade (chamada probabilidade de mutação Sempre Melhor) a qual dos dois tipos cada um dos filhos será submetido, sendo que esta escolha é feita individualmente. Assim, um filho pode ser submetido a um tipo de mutação e o outro filho, ao outro tipo. Alguns experimentos serão realizados com o objetivo

de avaliar o comportamento do AG no que tange à variação da probabilidade de mutação Sempre Melhor em relação à simples.

O primeiro tipo é chamado de mutação simples. Quando um indivíduo deve ser submetido a este tipo de mutação, cada gene existente no cromossomo é testado individualmente, de acordo com a probabilidade de mutação, para verificar se o valor atual do gene deve ser alterado ou não. Caso deva ser modificado, o algoritmo escolhe, aleatoriamente, um dentre os três valores possíveis {*F*, *D*, *E*}. Caso contrário, o próximo gene é testado, e assim sucessivamente. Desta forma, pode acontecer que seja escolhido um valor idêntico ao que já existia. Esta é uma possibilidade interessante e dá certa flexibilidade ao algoritmo, pois permite que ele "se arrependa" de aplicar a mutação àquele gene, voltando atrás em sua decisão inicial.

A Figura 26 apresenta um exemplo de indivíduo gerado após a aplicação do operador de mutação simples.

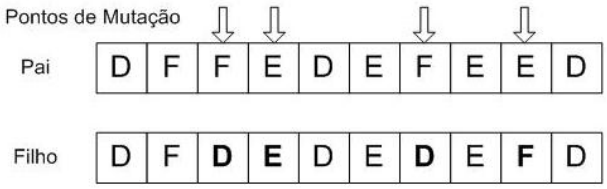


Figura 26: Aplicação de mutação simples.

O segundo tipo de mutação, denominado Mutação Sempre Melhor, funciona basicamente da mesma forma que a mutação simples. Cada gene é testado, de acordo com a mesma probabilidade de mutação, para decidir se deve ou não ter seu valor alterado. A única, e principal diferença, é que a cada mutação realizada, o indivíduo é avaliado novamente. Caso o *fitness* do indivíduo alterado seja maior do que o original, a mudança é mantida. Caso contrário, a alteração é restaurada e o procedimento prossegue com o próximo gene. Na maioria das vezes, este tipo especial de mutação consegue melhorar o *fitness* de um indivíduo, sendo que no pior caso, o *fitness* continuará como estava antes de ser submetido ao operador.

3.6 OPERADORES GENÉTICOS ESPECIAIS

A partir de alguns experimentos preliminares realizados, percebeu-se que somente a utilização dos operadores básicos não fornecia ao AG ajuda suficiente para que ele obtivesse

bons resultados. Assim, novos operadores genéticos foram desenvolvidos com o intuito de auxiliar o processo de evolução e obtenção de melhores resultados. Estes operadores estão descritos nas subseções a seguir.

3.6.1 U–*fold*

Após alguns experimentos preliminares, percebeu-se que algumas conformações encontradas no final da evolução do AG possuíam longas seqüências de resíduos dispostos em linha reta de modo a formarem contatos H–H com uma outra seqüência de resíduos em linha reta. Os algoritmos de aproximação descritos na seção 2.6.4 também utilizam este conceito para tentar simular o dobramento de uma proteína. A partir de então, decidiu-se implementar este operador.

Como o próprio nome indica, a idéia do operador U-*fold* é realizar um dobramento em forma de U de modo a maximizar o número de contatos entre as duas faces da seqüência que passam a ficar frente a frente. Inicialmente, verifica-se se o indivíduo selecionado deve ser submetido ao operador de acordo com um parâmetro chamado de probabilidade de U-*fold*.

Primeiramente, este operador varre todo o indivíduo (conformação) em busca do maior segmento do indivíduo que esteja em linha reta. A aplicação do operador somente prossegue caso o maior segmento em linha reta seja de, pelo menos, $1/10$ do tamanho total da proteína. Este valor foi escolhido empiricamente como sendo um valor razoável para que o operador possa participar ativamente do melhoramento dos indivíduos.

Após ter selecionado o maior segmento reto dentro do indivíduo, o operador testa todos os pontos de dobramento possíveis para aquele segmento para verificar qual é a melhor forma de dobrar este segmento de forma a maximizar o número de contatos H–H. Os dobramentos são testados para ambos os lados, esquerda e direita, conforme apresentado na Figura 27, para a seqüência HPHPPHHPHPPHPHHPPHPH. Para cada dobramento, o indivíduo é avaliado segundo seu *fitness*.

Ao final da aplicação deste operador, o melhor indivíduo encontrado é escolhido para fazer parte da população e prosseguir no processo de evolução. No caso do exemplo da Figura 27, o indivíduo escolhido é o número 2, já que este é o único indivíduo que possui 3 ligações hidrofóbicas não-locais.

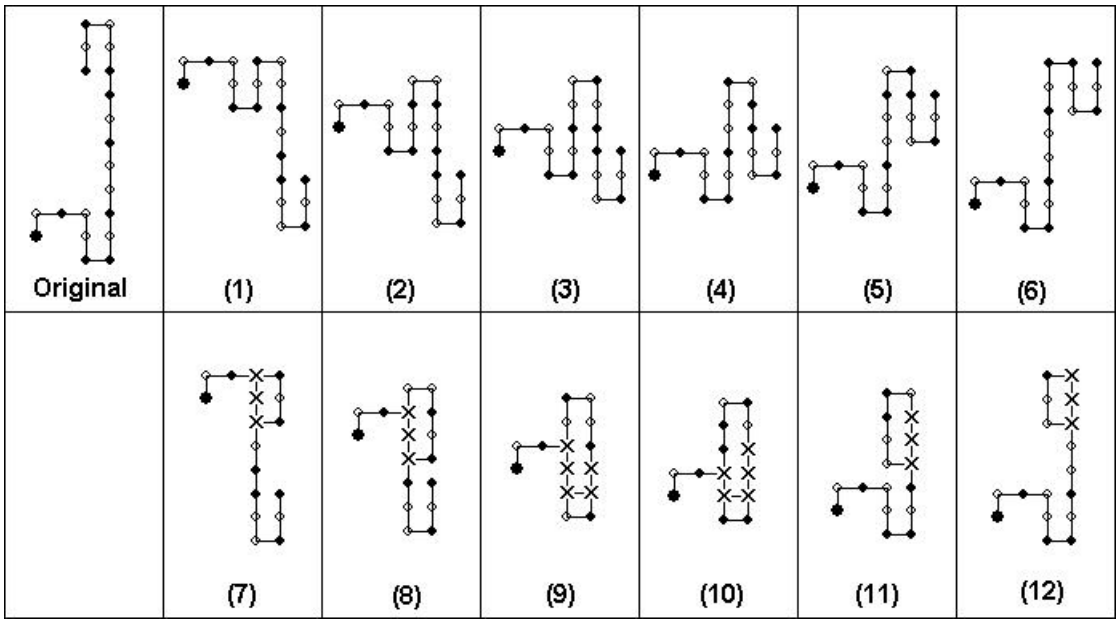


Figura 27: Possíveis dobramentos realizados pelo operador U-*Fold*.

Este operador necessita de segmentos retos relativamente grandes dentro das conformações. Entretanto, as conformações existentes vão se tornando mais compactas à medida que a evolução avança. Consequentemente, este operador só será útil nas primeiras gerações e não deveria continuar posteriormente. Desta forma, foi desenvolvida uma estratégia que faz com que este operador seja desabilitado automaticamente quando ele não estiver mais contribuindo para o processo evolutivo, economizando, assim, tempo computacional.

Cada vez que um indivíduo é submetido a este operador, um contador é atualizado. Se o indivíduo resultante do operador for melhor que o indivíduo original, o contador é incrementado em 1. Caso contrário, ele é decrementado de 1. Assim, quando, ao final da criação da população para a próxima geração, o valor deste contador for negativo (menor que zero), a probabilidade de aplicação deste operador nas próximas gerações passa a ser 0%, ou seja, ele é desabilitado. Deste modo, quando não for mais viável sua aplicação, o próprio AG o desabilita, evitando, assim, custo computacional adicional.

Isto é realizado a cada criação de uma nova população. Uma vez desabilitado, ele permanecerá desabilitado até o final da execução do AG. A única forma de habilitá-lo novamente durante a mesma execução em que foi desativado é quando acontece a dizimação da população (seção 3.7.1). Isto se justifica pelo fato de que novos indivíduos com poucas dobras são inseridos na população. Neste caso, o processo de avaliação da aplicação deste operador é novamente iniciado.

3.6.2 Gera *loops*

O operador Gera *loops* foi implementado com o objetivo de aumentar o número de contatos H–H de um indivíduo através do agrupamento de resíduos hidrofóbicos que se encontravam em linha reta em alguma conformação. Este operador também foi baseado nos conceitos apresentados na seção 2.6.4.

Da mesma forma como o operador U-*fold*, ele busca pelo maior segmento reto existente na conformação a ele submetida, sendo que este segmento precisa ser de, pelo menos, 4 resíduos de modo a possibilitar a aplicação do operador após submeter o indivíduo à probabilidade de Gera *loops*.

Após encontrar um segmento possível de ser aplicado, o operador busca, dentro deste segmento, o primeiro resíduo hidrofóbico. Ao encontrar, fixa esta posição e inicia uma segunda busca por outro resíduo hidrofóbico, sendo que este deve estar a, pelo menos, 4 resíduos de distância do primeiro. Além disto, para que um *loop* seja gerado, é necessário que exista um número par de resíduos entre o primeiro e o segundo resíduo hidrofóbicos. Esta limitação é devida ao tipo de grade que se está utilizando, uma grade quadrada.

Satisfeitas tais condições, o operador tenta gerar o *loop* posicionando os resíduos internos tanto para um lado quanto para o outro de forma a escolher o melhor tipo de *loop*, avaliando cada conformação gerada. Pode acontecer que ambas as conformações recém-geradas possuam *fitness* pior que o indivíduo original. Quando isto acontecer, o *loop* é desfeito e o indivíduo original, restaurado.

Caso o operador tenha sido aplicado e um *loop* tenha sido adicionado ao indivíduo, ele continua percorrendo o segmento reto encontrado para verificar se há a possibilidade de gerar um novo *loop*, mantendo o *loop* que foi adicionado. Isto é feito a partir da posição seguinte à do segundo resíduo selecionado anteriormente. Caso o indivíduo original tenha sido restaurado, o processo continua procurando o primeiro resíduo hidrofóbico existente a partir do primeiro resíduo selecionado anteriormente.

A Figura 28 apresenta a seqüência seguida pelo operador Gera *loops* para a proteína HPHPPHHPHHPHPHHPPHPH sendo que ao final do operador, a conformação escolhida é a número 4.

Da mesma maneira como o operador U-*fold* (seção 3.6.1), este não é muito utilizado quando a população já está bem evoluída, ou seja, as conformações já se encontram em algum estado razoavelmente compacto. Assim, o mesmo esquema usado para o operador U-*fold* foi

implementado para o Gera *loops*, permitindo que o AG possa desativá-lo automaticamente quando sua utilização passar a ser não muito adequada (recomendada).

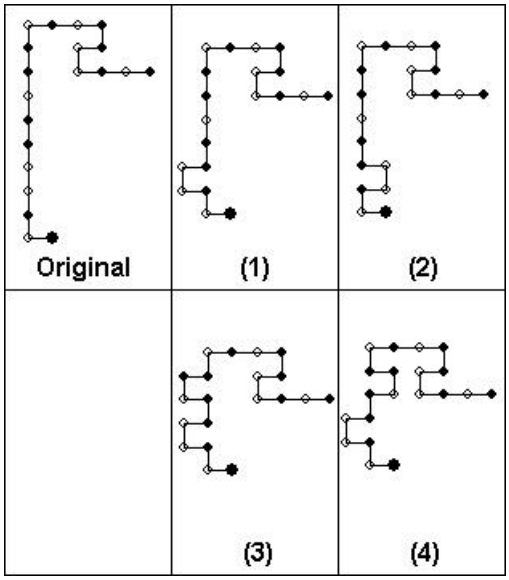


Figura 28: Exemplo de aplicação do operador Gera *loops*.

3.6.3 Otimização parcial

Este é um operador que realiza uma busca local e foi desenvolvido baseando-se em um conceito tirado de algoritmos para otimização de problemas combinatoriais, tal como o problema do caixeiro viajante (TSP – *Traveling Salesman Problem*), e é conhecido como 2-opt. O operador implementado é uma versão generalizada do conceito primeiramente proposto por (CROES, 1958).

Ao contrário dos demais operadores implementados, este não é realizado sobre a representação genotípica dos indivíduos (cromossomo), mas é aplicado ao fenótipo dos mesmos (posições dos resíduos na grade).

Primeiramente, são selecionadas aleatoriamente duas posições referentes a dois resíduos não consecutivos da proteína, fazendo com que suas posições permaneçam fixas na grade. Todos os resíduos anteriores à primeira posição selecionada e posteriores à segunda também são mantidos fixos na grade, de forma que somente as posições dos resíduos que se encontram entre as duas posições selecionadas possam ser alteradas. Este operador será aplicado somente à parte do indivíduo que não permaneceu fixa. A distância entre as duas posições selecionadas é chamada de "tamanho da otimização".

A partir de então, é aplicado um algoritmo que testa todos os possíveis dobramentos para o segmento interno não fixado. Para cada uma das possibilidades testadas, verifica-se se a conectividade da conformação como um todo foi mantida, pois muitas delas infringirão esta restrição. Para cada uma das possibilidades que respeitem esta restrição, o indivíduo completo é avaliado. Após todas as avaliações terem sido realizadas, a conformação que obteve o melhor *fitness* é selecionada para compor a população da próxima geração.

A Figura 29 mostra todas as possibilidades para um segmento de 3 resíduos de tamanho de otimização da proteína hipotética PHPHHPPHHP.

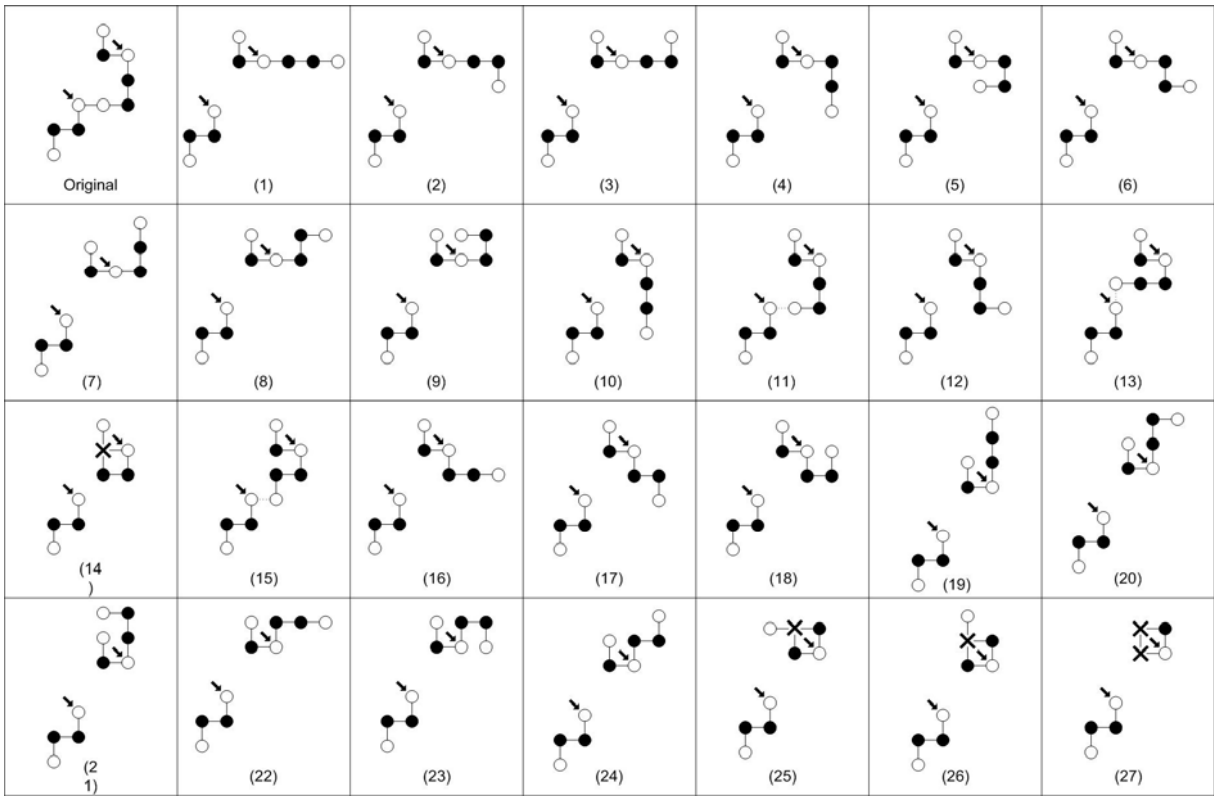


Figura 29: Todos os possíveis dobramentos para tamanho de otimização igual a 3.

Pode-se perceber que muitos dobramentos não mantêm a conectividade da cadeia, ou seja, somente as conformações 11, 13 e 15 representam conformações válidas e serão avaliadas. Como o operador testa todas as possibilidades, uma das conformações válidas sempre será o indivíduo original.

É fácil verificar que o número de possibilidades cresce exponencialmente à medida que o tamanho da otimização aumenta. Portanto, este operador deve ser utilizado com certo cuidado, pois dependendo do valor do parâmetro tamanho da otimização, o custo computacional da aplicação deste operador pode ser bastante elevado, inviabilizando sua

utilização. Também, este operador é aplicado probabilisticamente a todos os indivíduos selecionados para serem submetidos aos operadores genéticos de acordo com o parâmetro chamado probabilidade de otimização parcial.

Um exemplo de aplicação deste operador é apresentado na Figura 30, onde as setas indicam os resíduos que foram mantidos fixos. A Figura 30a mostra a conformação original com nenhum contato H–H e a Figura 30b apresenta a nova conformação, com duas ligações, após a aplicação do operador.

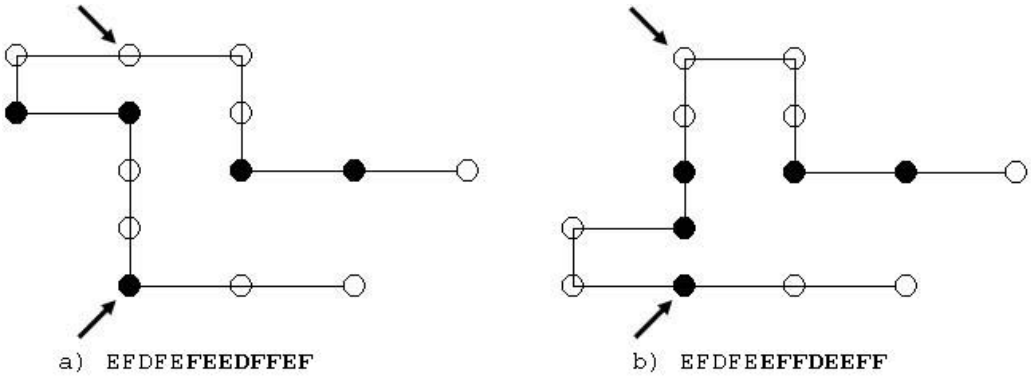


Figura 30: Aplicação da otimização parcial à seqüência PHHPPPPHHPPHPP.

Como os operadores genéticos especiais U-*fold* e Gera *loops* tendem a ser aplicados apenas nas primeiras gerações do AG, foi estabelecido que o operador de otimização parcial somente começasse a atuar a partir do momento em que ambos os operadores estivessem desativados. Ou seja, inicialmente, a probabilidade de aplicação do operador de otimização parcial é 0 (zero). A partir do momento que os outros operadores especiais forem desabilitados, a probabilidade de otimização parcial receberá o valor estabelecido pelo usuário. Esta estratégia foi estabelecida com o intuito de não prejudicar a aplicação dos operadores U-*fold* e Gera *loops*, já que estes se limitam às primeiras gerações.

3.7 NOVAS ESTRATÉGIAS

Nesta seção serão descritas algumas estratégias implementadas com o intuito de ajudar o AG com seus operadores genéticos a se comportar melhor durante o processo de evolução.

3.7.1 Dizimação

Testes preliminares mostraram que tanto a função de *fitness* proposta quanto os novos operadores genéticos implementados ajudaram o AG a alcançar um bom desempenho. Mesmo assim, é impossível assegurar que o AG não ficará estagnado em algum máximo local. Ao alcançar um máximo local, dependendo dos parâmetros selecionados, a diversidade genética da população pode ser rapidamente perdida. Este fenômeno, conhecido como convergência prematura, impede que possíveis melhoras futuras ocorram. Nesta situação, a evolução só continuaria caso a mutação fizesse com que outra parte do espaço de busca pudesse ser explorada, e mesmo assim, seria um evento do acaso. Contudo, é inútil permitir que o AG permaneça em um estado como este. Ao invés de encerrar sua execução e reiniciá-la, foi desenvolvida uma estratégia, chamada dizimação, descrita a seguir.

Durante todo o processo de evolução, o melhor indivíduo de cada geração é mantido. Caso após várias gerações o melhor indivíduo continuar sendo o mesmo, esta é uma grande evidência de que a população tenha possivelmente convergido e, consequentemente, a evolução tenha se estagnado. Esta estratégia verifica o *fitness* do melhor indivíduo da população durante a cada geração. Caso o *fitness* não tenha sido alterado, ou seja, o melhor indivíduo desta geração é o mesmo da geração anterior, um contador é incrementado em 1. Caso contrário, este mesmo contador é zerado. Assim, quando o contador atingir o valor 15, significando que durante as últimas quinze gerações o melhor indivíduo da população não evoluiu, a estratégia de dizimação é ativada.

Neste ponto do processo de evolução, antes de gerar a população da próxima geração, 80% desta população é dizimada, ou seja, os 80% piores indivíduos são eliminados da população. Os novos indivíduos serão criados da mesma forma como é gerada a população inicial e substituirão os piores que já foram eliminados. Este valor foi determinado após alguns experimentos preliminares e mostraram que para este valor de dizimação o algoritmo consegue obter bons resultados, aumentando a diversidade genética da população sem eliminar muito material genético que já tenha sido evoluído.

A aplicação da estratégia de dizimação faz com que a maior parte da população seja gerada novamente, resultando em uma grande diversidade genética. Além disto, ela permite que o processo evolutivo possa continuar por mais gerações. Porém, deve-se ater ao fato de que os indivíduos recém-criados terão provavelmente seus valores de *fitness* muito baixos, quando comparados aos indivíduos poupados da dizimação. Obviamente, torna-se necessário alterar os parâmetros atuais do AG de modo a diminuir a pressão seletiva (com relação aos

melhores indivíduos), de tal forma que todos os outros indivíduos possam ter a oportunidade de evoluir. Isto é atingido ao se diminuir o tamanho do torneio pela metade, permitindo que indivíduos piores também sejam selecionados e submetidos aos operadores genéticos. Também é necessário que os indivíduos possam ser melhorados mais rapidamente de modo a aumentar a média do *fitness* da população. Portanto, é dobrada a probabilidade de aplicar a mutação Sempre Melhor com o intuito de atingir este objetivo.

O efeito destas alterações nos parâmetros é uma rápida melhora no *fitness* médio da população em poucas gerações. Espera-se, com isto, que melhores indivíduos possam ser encontrados.

3.7.2 Melhorar última geração

Após testes com os operadores genéticos desenvolvidos e a estratégia de dizimação, achou-se interessante a utilização de uma última estratégia que pudesse trazer algum ganho à evolução.

Esta nova estratégia se baseia no fato de que, estando na última geração, não há mais nada a se fazer para se conseguir melhores indivíduos. Como uma última "cartada", esta estratégia consiste em aplicar o operador de mutação sempre melhor e de otimização parcial a todos os indivíduos selecionados que farão parte da última geração com o intuito de poder alcançar alguma melhora com relação aos indivíduos existentes. Ou seja, para gerar a população da última geração, a probabilidade de mutação sempre melhor e a probabilidade de otimização parcial são alteradas para 100%.

Embora seja computacionalmente custosa, esta estratégia de busca local é realizada apenas uma vez, tendo por objetivo um melhoramento da melhor solução encontrada pelo AG. A utilização desta estratégia é um parâmetro escolhido pelo usuário e, de modo geral, vem sempre a contribuir positivamente para o resultado final obtido.

3.7.3 Dobramento progressivo

Conforme apresentado na seção 2.3, à medida que as proteínas vão sendo geradas pelo ribossomo, o processo de dobramento já começa a acontecer, ou seja, após alguns resíduos já terem sido sintetizados, eles já começam a interagir com o meio e uns com os outros,

iniciando o dobramento. Entretanto, o dobramento estável somente acontece após todo um domínio ter sido gerado. Baseando-se neste fato, foi desenvolvida uma estratégia que busca simular o processo de dobramento de forma similar àquela que acontece na natureza, chamada Dobramento Progressivo.

A idéia básica desta estratégia é dividir a proteína em algumas partes e dobrar estas partes como se fossem domínios distintos. Para isto, ela permite que o usuário determine o número de partes (domínios) em que será dividida a proteína durante o processo de evolução. Cada uma das partes da proteína terá oportunidade de evoluir por uma quantidade igual de gerações.

A população inicial é gerada da mesma forma como explicado na seção 3.3, sendo que todos os indivíduos correspondem a codificações de apenas uma parte da proteína. Após evoluir por algumas gerações, uma segunda parte é acrescentada a todos os indivíduos da população. A codificação desta segunda parte corresponde a uma seqüência de Fs, ou seja, a todos os indivíduos é acrescentada uma quantidade de genes com valor F. Esta quantidade é determinada pelo número de resíduos que compõem a parte sendo acrescentada. O processo continua até que todas as partes tenham sido acrescentadas e evoluídas. Sendo assim, as últimas gerações evoluirão a proteína completa.

As partes são adicionadas na ordem inversa em que a proteína é formada. Por exemplo, se uma determinada proteína fosse dividida em 4 partes, a primeira parte a ser evoluída corresponderia aos últimos resíduos da proteína que pertencem à parte 4. Em seguida, seriam acrescentados os resíduos da parte 3. Prosseguindo a evolução, a segunda parte seria então adicionada, e, finalmente, a parte 1. Desta forma, esta seqüência tenta simular a ordem em que a proteína é sintetizada no ribossomo e a ordem em que as interações vão ocorrendo nos resíduos já sintetizados.

Para melhor ilustrar como este processo é realizado, a Figura 31 apresenta a seqüência de passos para a evolução da proteína HPHPPHHPHPPHPHHPPHPH de 20 resíduos sendo dividida em 4 partes. Apenas o melhor indivíduo de cada uma das 12 gerações é apresentado, sendo que a conformação 0 corresponde ao melhor indivíduo da população inicial.

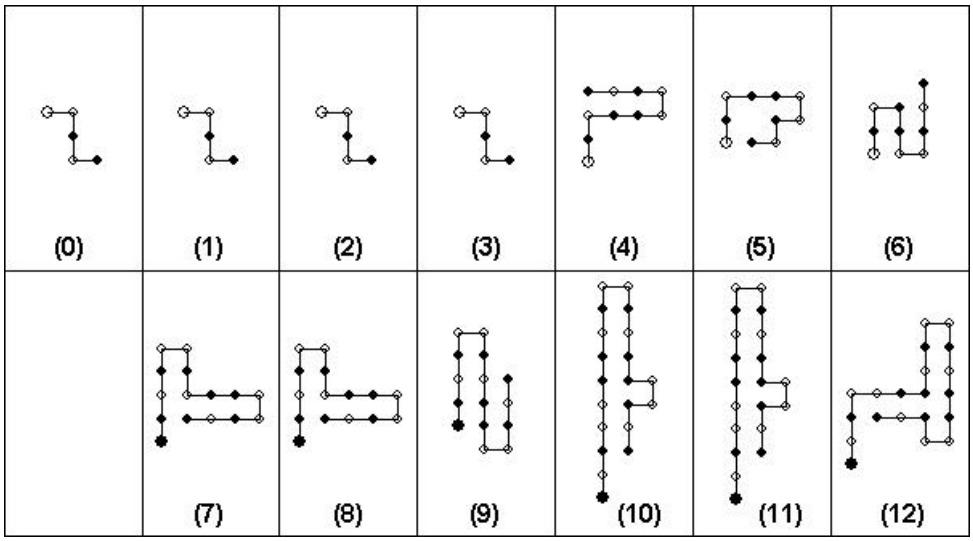


Figura 31: Exemplo de Dobramento Progressivo.

Analisando a Figura 31, percebe-se que a cada 3 gerações uma nova parte é acrescentada para que o processo de dobramento possa ser complementado até que a proteína esteja completa e, na última geração, o melhor indivíduo represente a melhor conformação obtida para a proteína. A primeira parte que foi dobrada, durante as três primeiras gerações, corresponde aos últimos 5 resíduos da proteína. Continuando, da geração 4 a 6, é acrescentada a parte imediatamente anterior aos últimos 5 resíduos, resultando em uma proteína com 10 resíduos e assim sucessivamente até que nas últimas 3 gerações o algoritmo considere a proteína completa permitindo, assim, que a evolução faça os “últimos ajustes” nas conformações existentes e alcance a melhor conformação possível.

3.8 FLUXO DE EXECUÇÃO DO AG

Após ter apresentado todas as características referentes ao algoritmo genético implementado, decidiu-se oferecer uma visão geral do funcionamento do AG, de forma que facilite a compreensão de como os operadores genéticos são aplicados e qual a relação entre eles. Não somente os operadores são explicados, mas também como as estratégias implementadas influenciam a execução do algoritmo e alteram os respectivos parâmetros.

A Figura 32 apresenta um diagrama de estados mostrando quais estados são percorridos pelo AG durante o processo de evolução. Isto é, a seqüência básica de execução do AG é apresentada. Foi utilizada a notação UML.

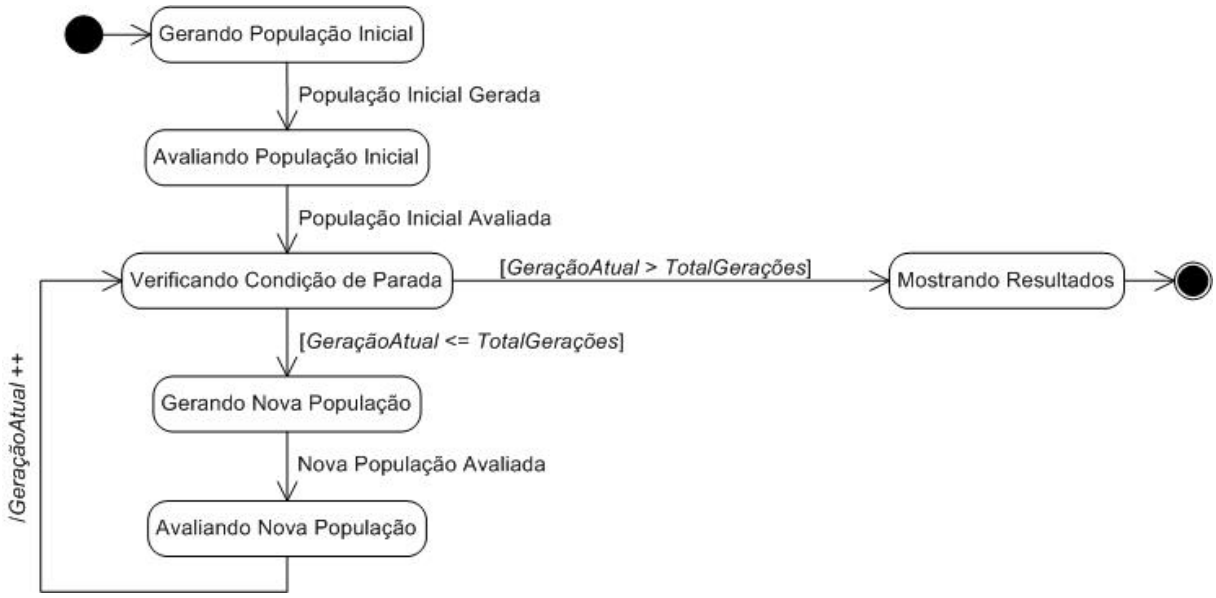


Figura 32: Diagrama de Estados do AG.

Inicialmente, a população inicial é gerada conforme descrito na seção 3.3. Estando pronta, esta população é submetida à função objetivo (de acordo com a seção 3.4) realizando uma avaliação dos indivíduos de modo a poder compará-los através de seus respectivos valores de *fitness*. O próximo estado que o AG passa é o que verifica a condição de parada do algoritmo (neste caso, a condição de parada é quando o número máximo de gerações estabelecidas tenha sido alcançado). Caso esta condição não tenha sido satisfeita, ou seja, $GeracaoAtual <= TotalGeracoes$, o algoritmo irá gerar uma nova população, sempre se baseando na população atual, a qual será avaliada a cada geração. Após a avaliação da população, o contador de gerações $GeracaoAtual$ será incrementado, sendo que a verificação da condição de parada deve ser feita logo após esta avaliação. Quando a condição for alcançada ($GeracaoAtual > TotalGeracoes$), os resultados são fornecidos ao usuário, ou seja, o melhor indivíduo de cada uma das gerações pelas quais o AG evoluiu é disponibilizado juntamente com os respectivos valores utilizados para o cálculo do *fitness*, inclusive.

De modo a esclarecer as interações decorrentes da aplicação dos operadores e das estratégias implementadas, a Figura 33 apresenta um diagrama que corresponde aos estados pelos quais o algoritmo passa durante o processo de geração de cada nova população.

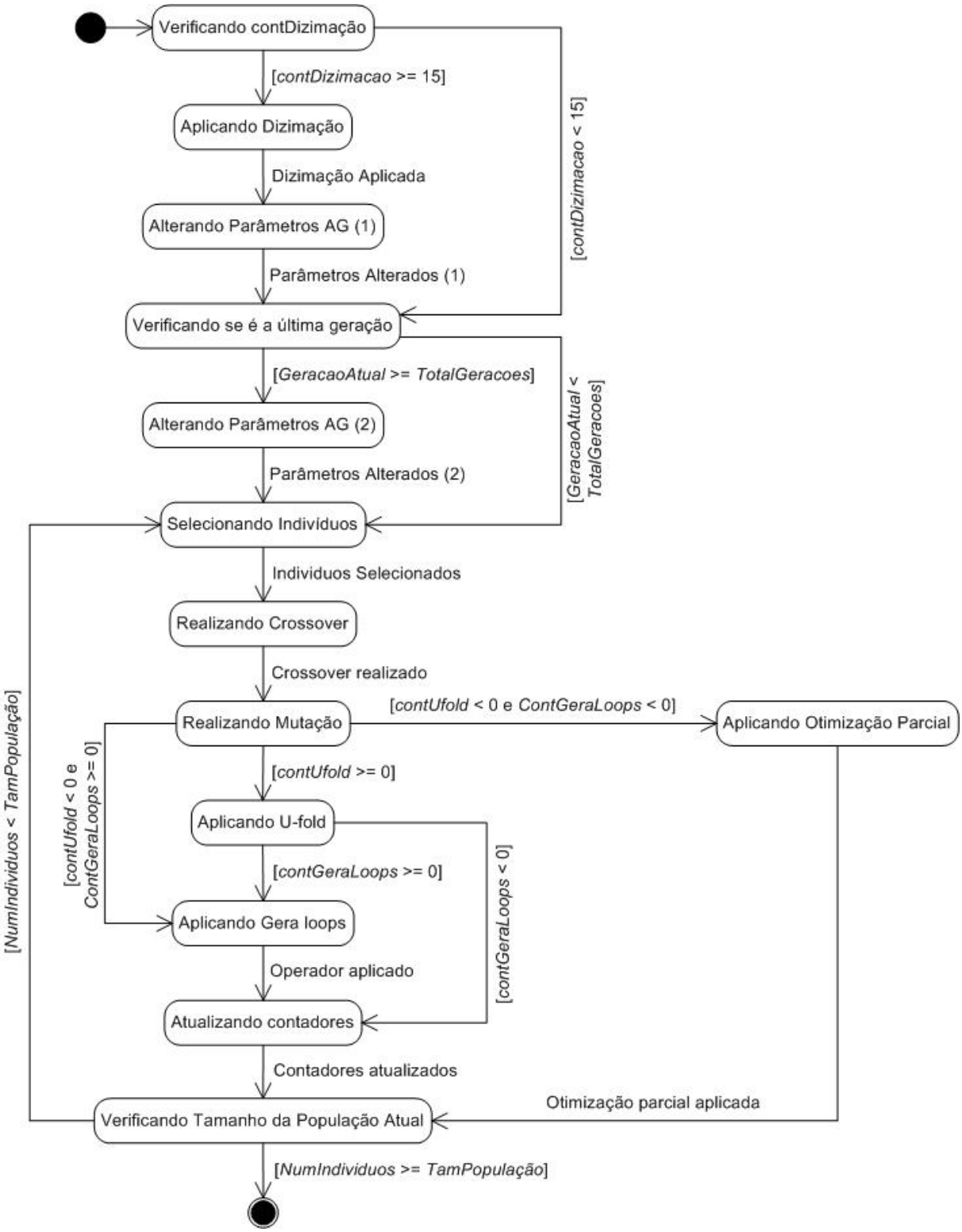


Figura 33: Detalhamento do estado “Gerando Nova População”.

O primeiro passo realizado na geração de uma nova população é verificar se será necessário aplicar a estratégia de dizimação a esta população. Caso a dizimação seja aplicada, o AG passa para o estado “Alterando Parâmetros AG (1)”, onde o tamanho do torneio e a

probabilidade de aplicação do operador Mutação Sempre Melhor são modificados conforme descrito na seção 3.7.1, passando em seguida para o estado “Verificando se é última geração”.

Caso a estratégia de dizimação não tenha sido aplicada, o AG passa para o estado “Verificando se é última geração”, onde é verificado se esta é a última geração. Caso afirmativo, há uma nova alteração de parâmetros antes do algoritmo prosseguir sua execução. Os parâmetros alterados no estado “Alterando Parâmetros AG (2)” são a probabilidade de mutação Sempre Melhor e a probabilidade de Otimização Parcial, de acordo com a seção 3.7.2. Caso esta estratégia não tenha sido aplicada ou após ter passado pelo estado “Alterando Parâmetros AG (2)”, o AG passa para a geração da população propriamente dita.

Para que uma nova população possa ser gerada, torna-se necessário selecionar probabilisticamente indivíduos da população atual para que os operadores genéticos possam ser a eles aplicados. Após a seleção dos mesmos, estes são submetidos ao operador de *crossover* (seção 3.5.1) e em seguida ao de mutação (seção 3.5.2).

O próximo passo da evolução é submeter os indivíduos selecionados aos operadores especialmente implementados para o problema. De acordo com a estratégia implementada e descrita nas seções 3.6.1 e 3.6.2, os contadores referentes a cada um dos operadores (U-*fold* e Gera *loops*, respectivamente) são testados. Quando ambos forem não negativos, ou seja, maiores ou iguais a zero, ambos os operadores são aplicados aos indivíduos selecionados na seqüência em que aparecem na Figura 33. Se ocorrer de apenas um deles não possuir valor negativo, somente este será aplicado. No caso de um destes operadores ter sido aplicado aos indivíduos, ou ambos, os respectivos contadores são atualizados para que seus valores possam ser usados na geração da próxima geração. Caso nenhum dos dois operadores anteriores tenha sido aplicado, os indivíduos são submetidos ao operador de Otimização Parcial, conforme detalhado na seção 3.6.3.

Após os indivíduos terem sido submetidos a todos os operadores genéticos, verifica-se quantos indivíduos já foram selecionados para fazerem parte da população da próxima geração. Se este número corresponder ao tamanho da população estabelecida pelo usuário, a geração da nova população encerra-se e o processo prossegue conforme a Figura 32. Caso a nova população ainda não esteja completa, novos indivíduos são selecionados e o processo de aplicação dos operadores é repetido.

Com relação à estratégia de Dobramento Progressivo o algoritmo segue seu fluxo normal, conforme já comentado previamente nesta seção. Porém, quando o número de partes em que a proteína deve ser dividida for maior do que 1, a estratégia é ativada. Inicialmente os indivíduos correspondem a uma parte da proteína, mas após um determinado número de

gerações o tamanho dos indivíduos é alterado para corresponder à quantidade de resíduos que estão sendo considerados naquele determinado momento da evolução. A evolução prossegue até o ponto em que a seqüência completa de resíduos é considerada. Para maiores detalhes, consultar a seção 3.7.3.

3.9 DESCRIÇÃO DO SISTEMA “GANDALF PRED”

O sistema “GANDALF PRED” foi implementado utilizando a ferramenta Borland Delphi 7 e se divide em 3 módulos: Parâmetros, Evolução e Resultados.

3.9.1 Módulo Parâmetros

Neste módulo do sistema pode-se acessar e modificar todos os parâmetros relacionados às características evolucionárias do AG. A tela referente a este módulo é apresentado na Figura 34.

Através desta tela é possível informar ao algoritmo qual proteína deverá ser dobrada (devendo seus resíduos estarem apropriadamente definidos no modelo HP) e o sistema calcula automaticamente o número de resíduos existentes. Os parâmetros referentes ao processo de evolução, tanto dos operadores genéticos básicos quanto dos operadores especiais, podem ser alterados nesta tela. Pode-se, ainda, habilitar e/ou desabilitar as estratégias desenvolvidas e selecionar o número de partes em que a proteína será dividida durante o processo de evolução.

Estando definidos os parâmetros desejados, é possível iniciar a evolução do AG clicando sobre o botão “Iniciar AG”. Caso seja desejado utilizar um conjunto pré-definido de parâmetros, pode-se clicar sobre o botão “Parâmetros *Default*”.

Outra possibilidade após a configuração dos parâmetros, é utilizar o processamento em lotes. Através desta opção, é possível determinar quantas vezes (rodadas) o AG repetirá o processo de evolução. Clicando sobre a opção “Salvar Automaticamente durante cada Rodada” faz com que o algoritmo salve um arquivo com o resultado parcial obtido até o momento da evolução após cada rodada que tenha acabado. Independente desta opção estar selecionada, o resultado final após todas as rodadas sempre será salvo no caminho previamente estabelecido.

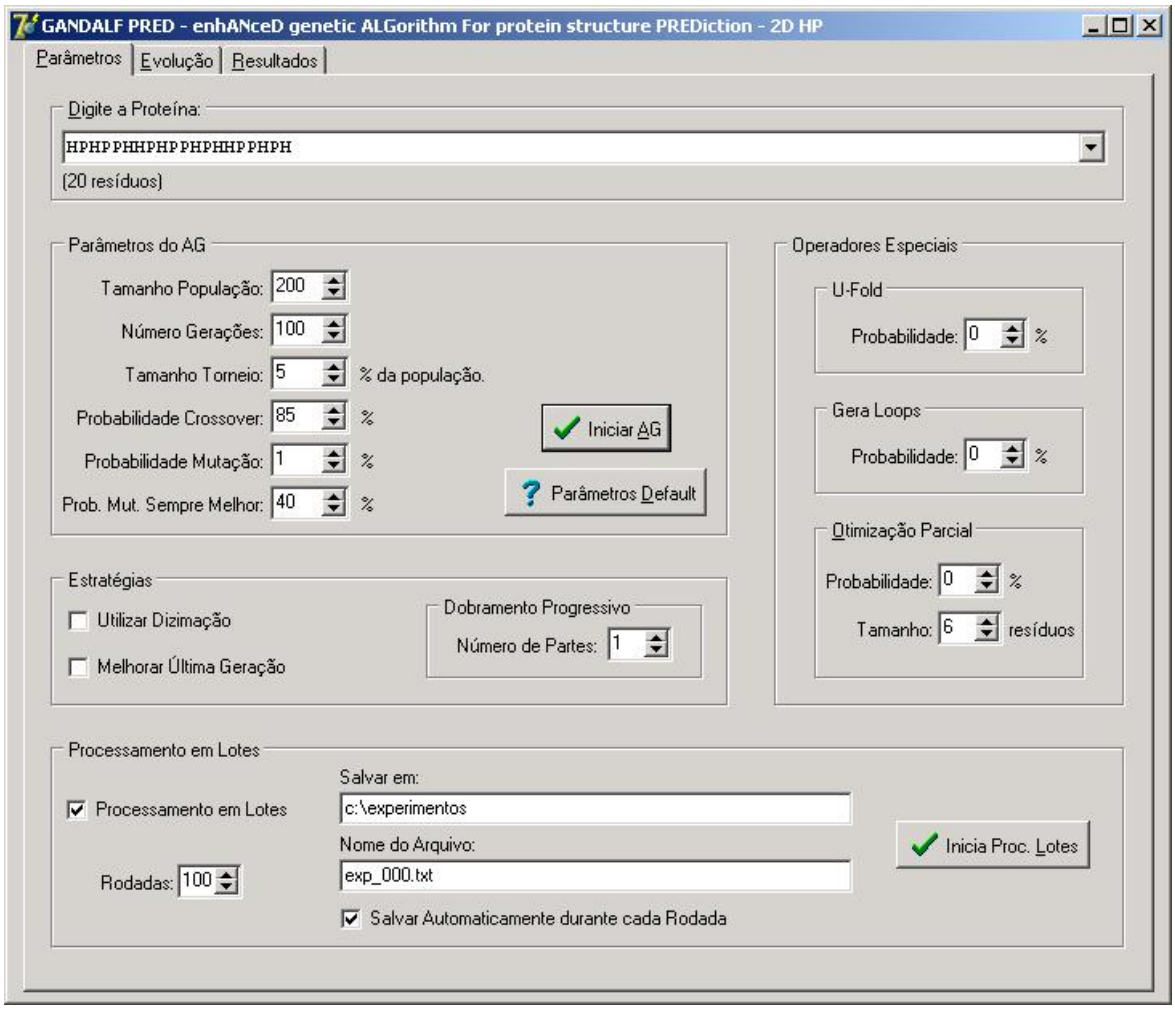


Figura 34: Configuração dos parâmetros do AG.

3.9.2 Módulo Evolução

Após iniciada a execução do AG, esta aba é automaticamente selecionada. Ela permite que o usuário acompanhe andamento do processo de evolução através da apresentação do "Status da Evolução" que mostra em qual parte do processo o algoritmo se encontra, além de apresentar qual a operação que está sendo realizada no momento.

Durante a evolução, é permitido ao usuário selecionar qual a forma de visualização do processo em execução que melhor lhe agrada. A troca de visualização entre as formas pode ser realizada durante o processo de evolução sem nenhum prejuízo à execução do algoritmo.

A primeira forma é o "Gráfico de Evolução". O gráfico de evolução do *fitness* permite que o usuário acompanhe o desenvolvimento dos indivíduos dentro da população. A cada geração este é atualizado dinamicamente apresentando o *fitness* máximo, médio e mínimo da população até a geração corrente. Este recurso está disponível a qualquer momento bastando

apenas que a opção “Mostrar Gráfico Evolução” esteja marcada. A Figura 35 apresenta um gráfico de evolução do *fitness* projetado sobre 50 gerações.

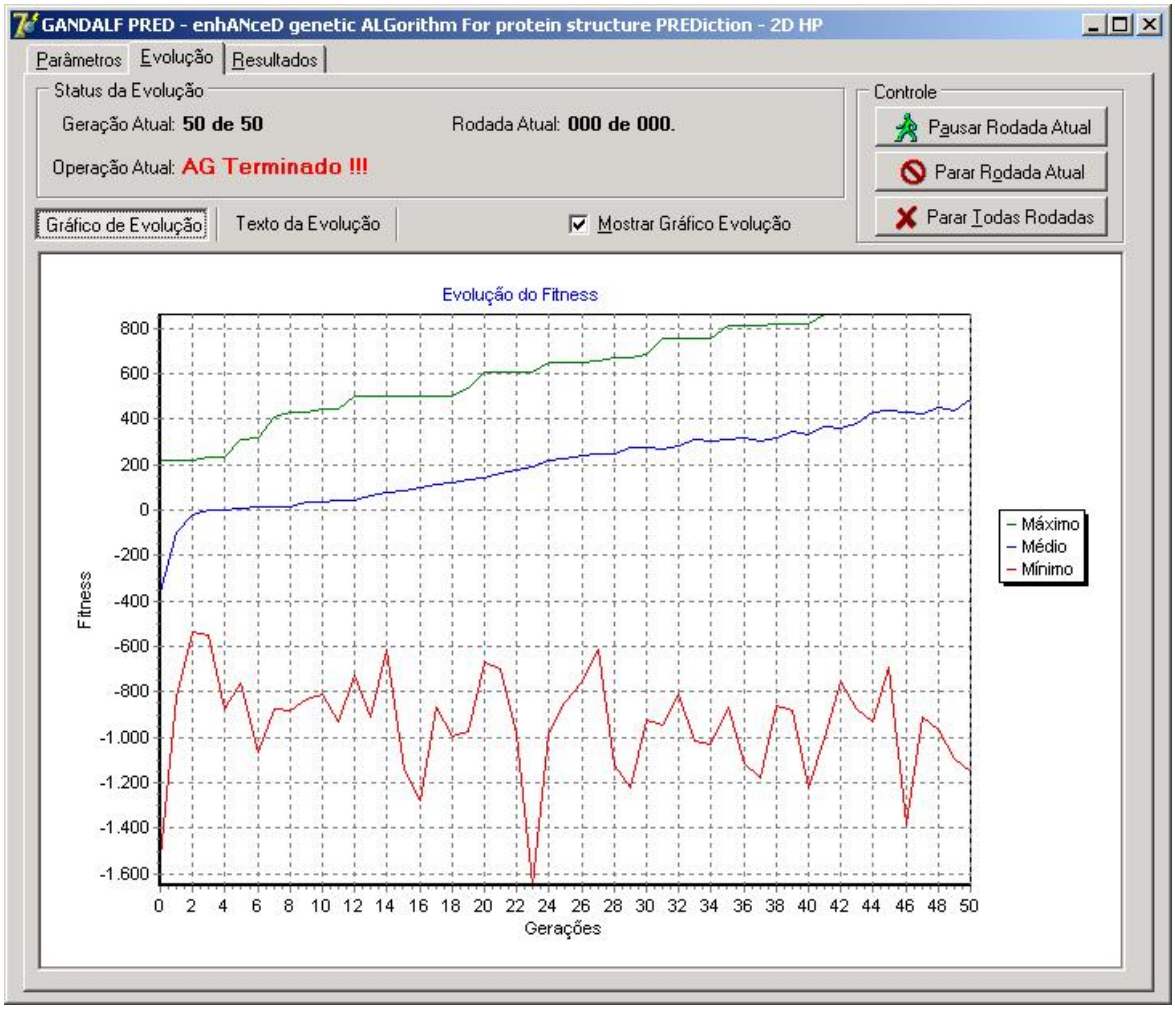


Figura 35: Gráfico de evolução do *fitness*.

A segunda forma de visualizar os resultados é fornecida em modo texto através da opção “Texto da Evolução”, independentemente da evolução estar sendo processada em lotes ou não, e encontra-se ilustrada na Figura 36 para uma execução individual.

Primeiramente, são fornecidos todos os parâmetros definidos pelo usuário que serão utilizados durante o processo de evolução. Logo em seguida, são apresentadas informações referentes ao melhor indivíduo da geração atual (ou ao melhor da rodada atual, no caso de processamento em lotes).

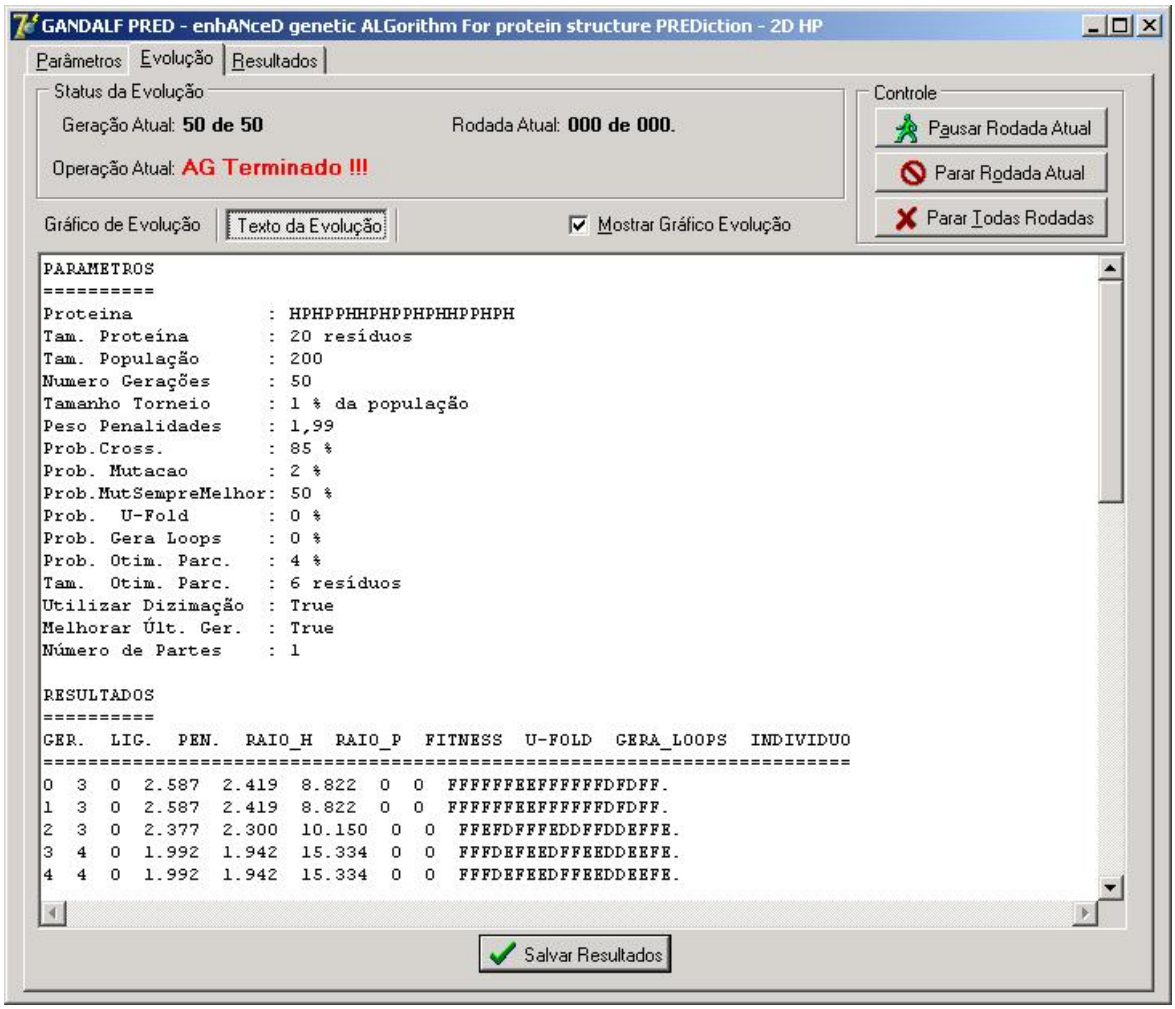


Figura 36: Texto da evolução do *fitness* (Execução Individual).

Caso o usuário tenha decidido executar o algoritmo por apenas uma rodada, as informações fornecidas correspondem às seguintes colunas:

- GER: geração atual;
- LIG: número de ligações hidrofóbicas não-locais;
- PEN: quantidade de colisões existentes no indivíduo;
- RAIO_H: valor do raio de giração dos resíduos hidrofóbicos;
- RAIO_P: valor do raio de giração dos resíduos hidrofílicos;
- FITNESS: valor final do *fitness* do indivíduo;
- U-FOLD: valor do contador do operador U-*fold* para a geração atual;
- GERA_LOOPS: valor do contador do operador Gera *Loops* para a geração atual;
- INDIVÍDUO: cromossomo do melhor indivíduo da geração atual.

Se o usuário resolver utilizar o processamento em lotes, as informações mostradas correspondem ao melhor indivíduo de uma rodada individual, e são:

- ROD: rodada atual;
- LIG: número de ligações hidrofóbicas não-locais;
- PEN: quantidade de colisões existentes no indivíduo;
- RAIO_H: valor do raio de giração dos resíduos hidrofóbicos;
- RAIO_P: valor do raio de giração dos resíduos hidrofílicos;
- FITNESS: valor final do *fitness* do indivíduo;
- GERAÇÃO: geração na qual foi encontrada o melhor indivíduo;
- DURAÇÃO: tempo despendido para evoluir uma determinada rodada, em segundos;
- U-FOLD: por quantas gerações o operador foi aplicado antes de ser desabilitado;
- GERA_LOOPS: número de gerações que o operador foi aplicado antes de ter sido desabilitado;
- INDIVÍDUO: cromossomo do melhor indivíduo da rodada atual.

Quando se utiliza o processamento em lotes, ao final de todas as rodadas é, ainda, apresentado um conjunto de valores correspondentes ao total de rodadas realizadas sob o nome de “Estatísticas”. São eles:

- Média de Ligações: média da coluna LIG;
- Desv. Pad. Ligações: desvio padrão da coluna LIG;
- Média de Gerações: média da coluna GERAÇÃO;
- Desv. Pad. Gerações: desvio padrão da coluna GERAÇÃO;
- Máx. de Ligações: número de ligações do melhor indivíduo de todas as rodadas e quantas vezes este valor foi atingido;
- Tempo Médio: média da coluna DURAÇÃO;
- Média U-FOLDs: média da coluna U-FOLD;
- Média Gera Loops: média da coluna GERA_LOOPS;

Além destas opções de visualização, é possível controlar a execução do algoritmo através dos botões de controle, podendo-se pausar/reiniciar uma determinada rodada, parar a rodada atual e encerrar a evolução, parando todas as rodadas subseqüentes.

3.9.3 Módulo Resultados

Quando o processo de evolução termina, o usuário pode selecionar esta aba para que os resultados obtidos pela execução do algoritmo possam ser visualizados de forma gráfica, conforme exemplo mostrado na Figura 37.

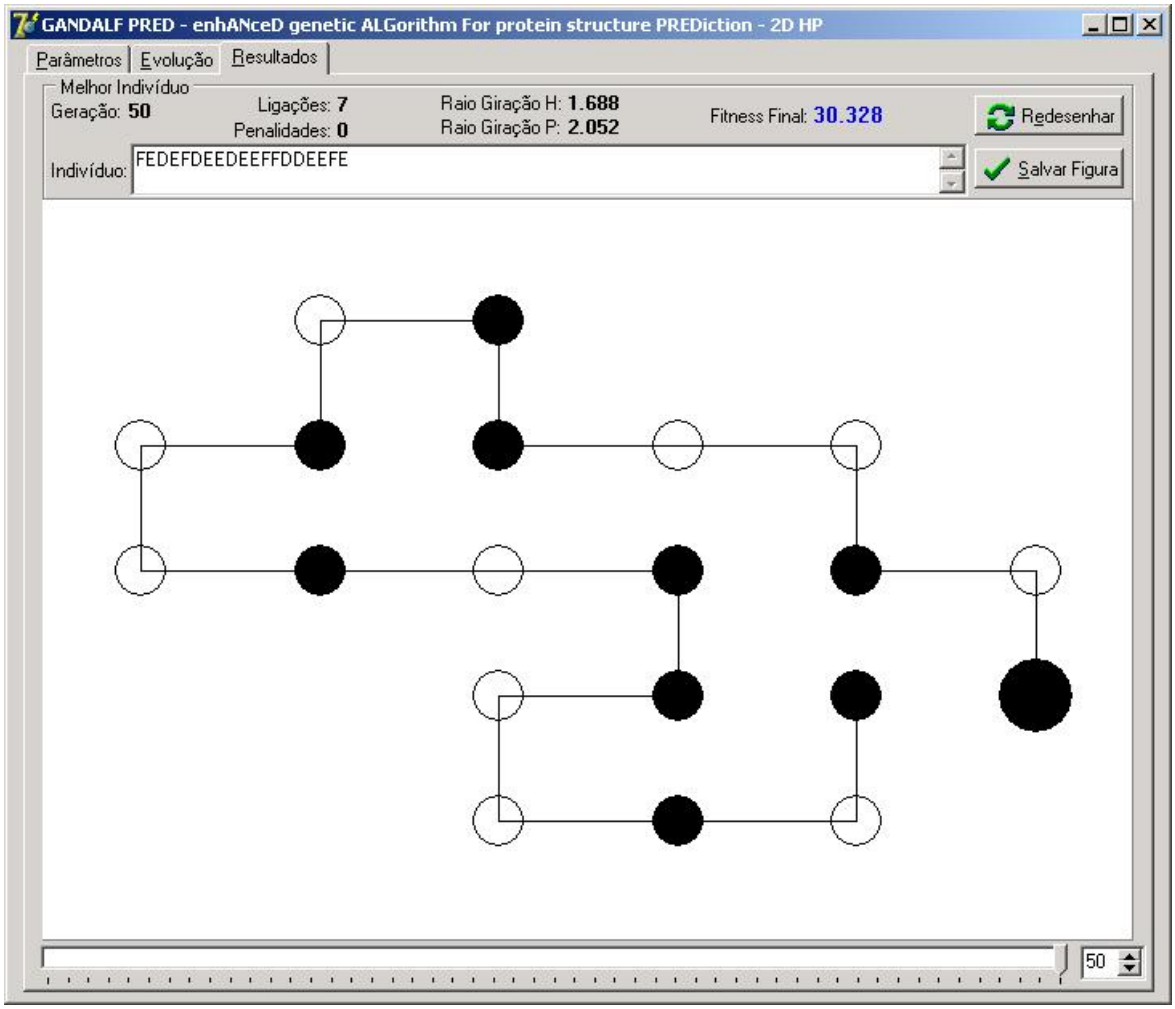


Figura 37: Visualização dos resultados.

Na parte superior são apresentadas informações referentes ao indivíduo que está sendo exibido, inclusive seu próprio cromossomo. É possível, ainda, alterar algum gene do indivíduo e visualizar a alteração realizada clicando sobre o botão “Redesenhar”, permitindo-se salvar o desenho do indivíduo após cada alteração.

Na parte inferior da aba é possível selecionar o melhor indivíduo de uma determinada geração (no caso de uma execução individual) ou o melhor indivíduo de uma determinada rodada, no caso de processamento em lotes. Para isto basta apenas informar o número da geração/rodada do indivíduo que se deseja visualizar.

CAPÍTULO 4

EXPERIMENTOS E RESULTADOS

4.1 DEFINIÇÃO DE PARÂMETROS DO AG

Antes de utilizar um AG para uma determinada classe de problemas, é interessante encontrar um conjunto de parâmetros que supostamente obtenha bons resultados para uma certa faixa de instâncias do problema. Porém, isto pode não ser possível devido à complexidade do *fitness landscape*. Este estudo não deve considerar somente o desempenho do algoritmo referente a seu resultado final, mas também com relação ao esforço computacional despendido.

O algoritmo genético proposto no Capítulo 3 oferece uma série de parâmetros e estratégias que podem alterar significativamente o desempenho do algoritmo. Estes parâmetros serão variados em determinadas faixas de valores e sua influência na evolução, bem como o resultado final, serão analisados.

Para a realização destes experimentos foi escolhida uma proteína retirada do PDB (*Protein Data Bank*) (BERMAN, WESTBROOK, FENG *et al*., 2000) chamada "*The Crystal Structure Of Fibroblast Growth Factor 7/1 Chimera*" cujo código de identificação é 1QQL. Ela possui 140 resíduos e foi escolhida devido à sua conformação tridimensional ser bastante compacta de modo a refletir os conceitos utilizados no cálculo da função objetivo. Como esta proteína foi retirada do PDB, não se tem conhecimento do número máximo de ligações hidrofóbicas não-locais que ela possui no modelo HP. No entanto, a maior seqüência, em que o número máximo de ligações é conhecido, possui apenas 85 resíduos. Por ser pequena, testes preliminares mostraram que ela não apresentava dificuldades para o algoritmo, sendo que variações nos parâmetros não representavam ganho algum aos resultados já obtidos. Assim, com uma proteína maior supõe-se que o algoritmo terá mais possibilidades de expressar as diferenças representadas pelas variações nos parâmetros que serão testados.

Desta forma, supõe-se que os ajustes nos parâmetros a serem realizados para esta proteína possam apresentar resultados similares quando aplicados a outras proteínas com estruturas similares.

Devido à natureza estocástica do AG, todos os experimentos descritos nas próximas seções foram realizados 100 vezes, sendo apresentadas apenas as médias referentes aos resultados. Os experimentos foram realizados em computadores Athlon XP-2.4 com 512 MB de memória RAM rodando Microsoft Windows 2000 Server.

É importante ressaltar que a Mutação Sempre Melhor nunca é utilizada nos experimentos, exceto quando estritamente definido, de forma a evitar que o algoritmo seja influenciado por algum operador de busca local.

4.1.1 Parâmetros básicos do AG

Inicialmente, os parâmetros básicos do AG foram submetidos a alguns testes com o intuito de tentar estabelecer um conjunto de valores para os mesmos. A seguir são listados os parâmetros testados com seus respectivos valores:

- Tamanho da População (*TamPop*), assumindo os valores 200 e 500;
- Número de Gerações (*NumGer*), assumindo os valores 100, 200 e 300;
- Tamanho do Torneio (*TamTorneio*), assumindo os valores 3% e 5%;
- Probabilidade de *Crossover* (*ProbCross*), assumindo os valores 70% e 90%;
- Probabilidade de Mutação (*ProbMut*), assumindo os valores 2%, 5% e 8%.

A partir deste conjunto de valores, foram realizados 72 experimentos com todas as combinações possíveis dos valores dos parâmetros. Os resultados foram computados em função da média de ligações H–H, média de gerações (representando em qual geração, em média, o melhor indivíduo foi encontrado), máximo de ligações (e o número de vezes que este valor foi encontrado) e tempo de processamento, conforme visto na seção 3.9.2. A Tabela 4 apresenta os resultados obtidos.

Como se pode observar na Tabela 4, a diferença dos valores da coluna “média de ligações” para os 10 melhores conjuntos de parâmetros é muito pequena (experimentos 64, 52, 61, 49, 40, 37, 34, 70, 67 e 58). Também, o número máximo de ligações encontradas pelos melhores experimentos mostrados na tabela não difere muito, mas como a diferença de uma ligação entre duas conformações corresponde a um grande avanço alcançado pelo algoritmo, inicialmente, somente os experimentos 64 e 52 foram considerados. Com relação ao tempo de

processamento para estes dois experimentos, a diferença foi de apenas alguns segundos, não sendo, portanto, significativa de forma a impactar o rendimento do algoritmo.

Desta forma, decidiu-se por escolher o conjunto de parâmetros correspondentes ao experimento 64 como o conjunto inicial padrão para estes parâmetros, já que possui a média de ligações maior do que os outros experimentos. Os valores correspondentes aos parâmetros utilizados neste experimento são: $TamPop = 500$, $NumGer = 300$, $TamTorn = 3\%$, $ProbCross = 90\%$ e $ProbMut = 2\%$.

Estes valores foram fixados para serem usados na execução dos demais experimentos onde os parâmetros dos operadores especiais foram avaliados.

Tabela 4: Resultados dos testes para estabelecer valores para os parâmetros básicos.

Exp.	Tam Pop	Num Ger	Tam Torn	Prob Cross	Prob Mut	Média Ligações	Média Gerações	Máx. Ligações	Tempo (s)
1	200	100	3	70	2	34,72	84,11	44(1x)	3,80
2	200	100	3	70	5	18,93	72,29	25(4x)	3,84
3	200	100	3	70	8	14,56	56,46	25(1x)	3,80
4	200	100	3	90	2	35,31	87,56	45(3x)	4,00
5	200	100	3	90	5	18,30	65,62	25(3x)	4,05
6	200	100	3	90	8	14,52	56,91	25(1x)	4,00
7	200	100	5	70	2	44,06	86,72	55(1x)	3,66
8	200	100	5	70	5	22,70	74,94	35(1x)	3,80
9	200	100	5	70	8	16,79	62,23	26(1x)	3,78
10	200	100	5	90	2	44,82	85,51	53(1x)	3,87
11	200	100	5	90	5	22,99	70,26	32(1x)	4,00
12	200	100	5	90	8	16,69	56,56	25(1x)	3,99
13	200	200	3	70	2	38,79	164,12	52(1x)	7,72
14	200	200	3	70	5	20,89	145,98	31(1x)	7,83
15	200	200	3	70	8	17,37	123,75	28(1x)	7,72
16	200	200	3	90	2	38,55	168,34	47(1x)	8,14
17	200	200	3	90	5	20,55	144,15	29(1x)	8,24
18	200	200	3	90	8	15,22	123,48	22(1x)	8,19
19	200	200	5	70	2	45,64	161,17	56(2x)	7,47

Tabela 4: Resultados dos testes para estabelecer valores para os parâmetros básicos. (cont.)

Exp.	Tam Pop	Num Ger	Tam Torn	Prob Cross	Prob Mut	Média Ligações	Média Gerações	Máx. Ligações	Tempo (s)
20	200	200	5	70	5	25,27	143,91	36(1x)	7,76
21	200	200	5	70	8	18,04	128,61	27(2x)	7,73
22	200	200	5	90	2	46,77	155,94	56(1x)	7,87
23	200	200	5	90	5	24,94	148,91	33(1x)	8,21
24	200	200	5	90	8	17,75	120,08	28(1x)	8,18
25	200	300	3	70	2	40,25	247,79	48(2x)	11,51
26	200	300	3	70	5	22,12	225,78	31(2x)	11,65
27	200	300	3	70	8	17,16	198,50	26(1x)	11,49
28	200	300	3	90	2	40,81	253,90	48(2x)	12,13
29	200	300	3	90	5	21,36	219,08	30(1x)	12,27
30	200	300	3	90	8	17,14	186,60	27(1x)	12,30
31	200	300	5	70	2	47,65	243,50	57(1x)	11,08
32	200	300	5	70	5	26,99	235,44	36(1x)	11,57
33	200	300	5	70	8	19,89	220,86	28(1x)	11,51
34	200	300	5	90	2	48,69	232,22	57(1x)	11,70
35	200	300	5	90	5	26,32	225,64	37(1x)	12,25
36	200	300	5	90	8	19,53	199,96	30(1x)	12,14
37	500	100	3	70	2	48,74	77,89	56(1x)	10,74
38	500	100	3	70	5	28,07	75,98	38(1x)	11,27
39	500	100	3	70	8	19,51	63,70	26(2x)	11,26
40	500	100	3	90	2	48,94	76,09	56(2x)	11,25
41	500	100	3	90	5	28,23	72,63	34(3x)	11,77
42	500	100	3	90	8	19,24	61,84	28(1x)	11,76
43	500	100	5	70	2	46,81	66,23	56(2x)	10,61
44	500	100	5	70	5	34,70	78,47	44(1x)	11,14
45	500	100	5	70	8	22,87	64,82	30(1x)	11,23
46	500	100	5	90	2	46,95	66,95	55(1x)	11,12
47	500	100	5	90	5	35,09	75,01	42(1x)	11,61
48	500	100	5	90	8	22,01	60,54	30(1x)	11,77

Tabela 4: Resultados dos testes para estabelecer valores para os parâmetros básicos. (cont.)

Exp.	Tam Pop	Num Ger	Tam Torn	Prob Cross	Prob Mut	Média Ligações	Média Gerações	Máx. Ligações	Tempo (s)
49	500	200	3	70	2	49,42	148,93	57(1x)	21,23
50	500	200	3	70	5	30,66	155,98	38(1x)	22,48
51	500	200	3	70	8	20,99	127,45	30(1x)	22,47
52	500	200	3	90	2	50,62	146,16	59(1x)	22,27
53	500	200	3	90	5	30,73	149,00	40(1x)	23,56
54	500	200	3	90	8	21,10	136,56	31(1x)	23,52
55	500	200	5	70	2	47,24	139,07	57(1x)	21,11
56	500	200	5	70	5	36,26	160,61	44(2x)	22,24
57	500	200	5	70	8	24,17	134,08	34(1x)	22,48
58	500	200	5	90	2	48,17	135,39	55(3x)	22,09
59	500	200	5	90	5	37,76	160,92	46(1x)	23,24
60	500	200	5	90	8	23,72	129,81	33(1x)	23,61
61	500	300	3	70	2	50,15	208,43	58(1x)	31,83
62	500	300	3	70	5	31,58	231,26	39(2x)	33,57
63	500	300	3	70	8	22,20	190,98	36(1x)	33,64
64	500	300	3	90	2	50,91	226,16	59(1x)	33,46
65	500	300	3	90	5	31,55	220,84	38(2x)	35,23
66	500	300	3	90	8	21,92	193,16	32(1x)	35,22
67	500	300	5	70	2	48,24	207,93	58(1x)	31,49
68	500	300	5	70	5	38,80	234,31	48(2x)	33,37
69	500	300	5	70	8	25,38	211,09	33(1x)	33,63
70	500	300	5	90	2	48,38	205,90	57(1x)	33,13
71	500	300	5	90	5	39,09	237,65	49(1x)	35,02
72	500	300	5	90	8	24,83	199,20	33(1x)	35,32

4.1.2 Probabilidade da Mutação Sempre Melhor

Após escolher um conjunto de valores para os parâmetros básicos do algoritmo genético, é importante avaliar o comportamento dos novos operadores implementados com o intuito de avaliar seu impacto no desempenho.

Assim, o primeiro operador especial a ser testado é a Mutação Sempre Melhor. Conforme apresentado na seção 3.5.2, antes do operador de mutação ser aplicado a um indivíduo, é necessário selecionar qual o tipo de mutação que será aplicada, a simples ou a que sempre melhora. Esta seleção é realizada através de um parâmetro que indica a probabilidade de um indivíduo ser submetido à Mutação Sempre Melhor, chamado *ProbMutSempreMelhor*.

De forma a avaliar a influência deste parâmetro no processo de evolução, foram realizados alguns experimentos. Para isto, foram testados os valores {10, 30, 50, 70, 90, 100}, em porcentagem. Os resultados obtidos pelos experimentos realizados estão sumarizados na Tabela 5.

Tabela 5: Resultados de *ProbMutSempreMelhor*.

Exp.	***ProbMut SempreMelhor***	**Média Ligações**	**Média Gerações**	**Máx. Ligações**	**Tempo (s)**
64	0	50,91	226,16	59(1x)	33,46
73	10	50,28	193,92	59(1x)	37,51
74	30	48,09	189,90	57(1x)	44,31
75	50	48,36	185,28	58(2x)	44,97
76	70	47,37	171,53	58(1x)	56,49
77	90	46,39	134,37	57(1x)	61,40
78	100	45,62	12,64	54(2x)	56,40

De acordo com a Tabela 5, percebe-se que o algoritmo consegue obter melhores resultados quando pouca Mutação Sempre Melhor é utilizada. O objetivo do operador de mutação em um AG é manter a diversidade genética da população, de modo a evitar a convergência prematura. Com mais mutação simples sendo aplicada, o algoritmo mantém esta diversidade por mais tempo. À medida que o valor de *ProbMutSempreMelhor* vai

aumentando, o algoritmo converge mais prematuramente, conforme pode ser verificado pelos valores da coluna “Média Gerações”, resultando, normalmente, em resultados inferiores.

Porém, o melhor resultado obtido dentre todos os experimentos realizados foi quando o operador não foi utilizado (experimento 64), o que não era esperado. Provavelmente, isto reflete o fato de o algoritmo perder diversidade genética quando a mutação deixa de ser probabilística. Um exemplo claro disto é o experimento 78, que mostra um exemplo de convergência prematura, já que, em média, o melhor indivíduo é encontrado por volta da geração 12.

Portanto, para o parâmetro *ProbMutSempreMelhor*, o valor a ser considerado é de 0%. Porém, poder-se-ia definir o valor de *ProbMutSempreMelhor* como 10% (experimento 73) já que sua diferença com relação à média de ligações do experimento 64 não é significativa. No entanto, isto não foi feito haja vista que a média de ligações corresponde à média de 100 rodadas, o que confere ao valor da média um certo grau de confiança.

4.1.3 Probabilidade de U-*Fold*

Para testar a influência deste operador na evolução e estabelecer um valor pré-definido para sua probabilidade de aplicação (*ProbUFold*), foram realizados experimentos com os valores {10, 20, 30, 40, 50, 60, 70, 80, 90, 100}, em porcentagem, a partir dos parâmetros definidos na seção 4.1.1. Os resultados obtidos pelos experimentos estão descritos na Tabela 6, sendo que a coluna “Média U-Fold” indica por quantas gerações, em média, o operador U-*fold* foi utilizado.

Observando-se a Tabela 6, percebe-se que, quando este operador é usado, em média, sempre se obtém um resultado melhor do que quando não é usado (experimento 64), independentemente do valor de sua probabilidade de aplicação. Isto sugere que este operador seja utilizado. Além disto, decidiu-se escolher como padrão para o parâmetro *ProbUFold* o valor 10% (experimento 79), de acordo com os comentários a seguir.

Apesar de não ter apresentado a melhor média de ligações entre todos os experimentos, sua diferença para o melhor é de apenas 0,35%. Além disso, este foi o valor que obteve a maior média de gerações. Também, apresentou o segundo melhor resultado em se tratando da quantidade máxima de ligações encontrada entre seus melhores indivíduos, ficando atrás apenas do valor encontrado pelo experimento 83.

Tabela 6: Resultados de *ProbUFold*.

Exp.	*ProbUFold*	**Média Ligações**	**Média Gerações**	**Máx. Ligações**	**Tempo (s)**	**Média U-Fold**
64	0	50,91	226,16	59(1x)	33,46	---
84	60	53,41	195,65	59(1x)	33,75	1,05
82	40	53,34	213,38	59(2x)	33,10	1,05
79	10	53,22	226,70	60(1x)	32,24	1,44
80	20	53,18	208,85	59(3x)	32,66	1,23
85	70	53,04	200,61	59(3x)	34,01	1,02
81	30	53,01	212,05	59(2x)	32,78	1,04
87	90	53,00	204,50	59(1x)	34,39	1,00
86	80	52,74	197,81	59(2x)	34,24	1,00
88	100	52,63	212,63	59(1x)	34,58	1,00
83	50	52,63	209,85	60(3x)	33,48	1,03

Com relação ao tempo, apesar da diferença não ter sido significativa entre todos os experimentos, o experimento 79 foi o que despendeu menos tempo, em média. Isto é importante, pois ao se utilizar vários operadores genéticos especiais ao mesmo tempo, esta pequena diferença pode se tornar bastante significativa com relação ao tempo de processamento necessário, principalmente quando uma proteína muito longa for dobrada. Finalmente, a última coluna, em média, indica por quantas gerações este operador foi aplicado durante a evolução antes de ser desabilitado. O experimento 79 apresentou o maior valor dentre todos os outros, significando que com o valor de 10%, o algoritmo maximiza o uso do operador.

Vale a pena lembrar que se o operador foi aplicado, em média, somente na primeira geração, então seu efeito foi sendo diluído ao longo das 299 gerações subseqüentes. Portanto, seu efeito só tem a ver com a região inicial do espaço de busca do AG.

4.1.4 Probabilidade de Gera *loops*

Os experimentos realizados para testar o valor da probabilidade de aplicar o operador Gera *loops* (*ProbGeraLoops*) utilizaram os valores percentuais de {10, 20, 30, 40, 50, 60, 70,

80, 90, 100}. A Tabela 7 apresenta os resultados obtidos, sendo que a coluna “Média Gera Loops” indica por quantas gerações, em média, o operador foi utilizado.

A escolha do valor 20% para o parâmetro *ProbGeraLoops* (experimento 90) foi feita a partir dos resultados obtidos. Segundo a Tabela 7, este experimento obteve a terceira melhor média de ligações além de ser o segundo em média de gerações. Com relação ao máximo de ligações encontradas, apesar do valor encontrado pelo experimento 89 ter sido significativamente maior do que os demais, verificou-se que o segundo melhor indivíduo encontrado por este experimento possuía apenas 59 ligações, ou seja, pode-se dizer que o fato dele ter encontrado um indivíduo com 65 ligações foi um pouco de “sorte”. Em outras palavras, como AGs são algoritmos probabilísticos, pode ter ocorrido de nesta rodada o algoritmo ter iniciado a busca em uma região do espaço de busca onde houvesse um máximo local um pouco melhor que em outras regiões, fazendo com que o melhor resultado apresentasse tal diferença. Sendo assim, as 60 ligações encontradas pelo experimento 90 são, neste caso, bastante significativas.

Tabela 7: Resultados para *ProbGeraLoops*.

Exp.	***Prob GeraLoops***	**Média Ligações**	**Média Gerações**	**Máx. Ligações**	**Tempo (s)**	**Média Gera Loops**
64	0	50,91	226,16	59(1x)	33,46	---
89	10	53,22	182,45	65(1x)	33,11	12,47
90	20	53,68	194,28	60(1x)	33,32	8,44
91	30	53,41	184,97	61(3x)	33,82	7,84
92	40	53,14	194,62	59(1x)	33,85	6,11
93	50	53,19	184,88	60(1x)	34,21	5,94
94	60	53,58	188,37	60(2x)	34,34	5,44
95	70	53,85	188,04	59(4x)	34,67	5,25
96	80	53,51	192,75	60(1x)	34,91	4,99
97	90	53,71	189,33	59(2x)	35,08	4,83
98	100	53,31	183,33	61(1x)	35,33	4,78

Observando o tempo decorrido, este é o segundo melhor valor, perdendo apenas para o experimento 89. Por fim, ao se analisar a média de gerações em que este operador foi aplicado, o valor 8,44 encontrado pelo experimento 90 é um valor razoável. Isto se deve ao

fato de que, quando da utilização de dizimação, este operador é habilitado novamente (caso não esteja). Lembrando que a dizimação da população ocorre após 15 gerações sem alguma melhora no *fitness* do melhor indivíduo, o valor 8,44 encontra-se aproximadamente na metade, ou seja, desta forma, quando todos os operadores genéticos especiais estão selecionados, o operador de otimização parcial terá aproximadamente 7 gerações para atuar antes que uma nova dizimação seja aplicada, devido à desabilitação dos operadores U-*fold* e Gera *loops*. Caso fosse escolhido um experimento cuja média de gerações de aplicação do operador Gera *loops* fosse maior, no exemplo citado acima, o operador de otimização parcial teria poucas gerações para atuar e tentar auxiliar a população recém-dizimada a melhorar.

Comparando-se os resultados com o apresentado pelo experimento 64, nota-se que para qualquer valor de probabilidade, o resultado obtido é sempre melhor do que quando o operador não é aplicado.

4.1.5 Probabilidade de otimização parcial

Para avaliar qual o comportamento do algoritmo quando o operador de Otimização Parcial é utilizado, foram realizados alguns experimentos nos quais os valores para os parâmetros *ProbOtimParcial* (probabilidade de otimização parcial) e *TamOtimParcial* (tamanho da otimização parcial) foram {1, 2, 3, 4, 5, 6, 7, 8} e {5, 6, 7, 8}, respectivamente. Os 32 experimentos realizados estão mostrados na Tabela 8.

Conforme comentado na seção 3.6.3, à medida que *TamOtimParcial* aumenta, o tempo de processamento também cresce de forma exponencial. Portanto, para a avaliação destes experimentos, além da média de ligações, também foi necessário levar em consideração o tempo de processamento do processo de evolução. Foi utilizado o conceito de "Otimalidade de Pareto" (DEB, 2001), sendo utilizada quando uma otimização multi-objetivos é necessária. A idéia básica deste procedimento é construir um gráfico onde cada eixo representa um dos fatores a serem analisados e cada ponto corresponde ao resultado obtido por cada combinação de parâmetros. Neste caso, existem apenas dois eixos, já que somente os fatores "média de ligações" e "tempo" estão sendo considerados. Cada ponto (x_i, y_i) é então classificado como sendo:

- Dominado, quando existe algum outro ponto (x_j, y_j) onde $(x_j < x_i) \wedge (y_j < y_i)$;
- Não-dominado, caso contrário.

Neste tipo de análise, somente os pontos não-dominados são considerados, pois eles representam um conjunto onde nenhum ponto é melhor do que o outro quando se consideram todos os objetivos ao mesmo tempo.

O eixo das abscissas indica o valor $(62 - \textit{médiaHnLB})$, onde *médiaHnLB* é o valor da coluna “Média Ligações” e o eixo das ordenadas apresenta os valores da coluna “Tempo”, ambos da Tabela 8. O valor 62 foi escolhido simplesmente por ser o menor valor inteiro maior que os valores encontrados entre as médias de ligações. O gráfico para este estudo é mostrado na Figura 38, onde os quadrados vazios representam os pontos dominados e os círculos preenchidos correspondem aos pontos não-dominados por nenhum outro ponto.

Como explicado anteriormente, esta análise não indica um ponto como o melhor conjunto de parâmetros, já que todos os pontos não-dominados representam soluções igualmente boas considerando-se ambos os critérios. Isto permite que o usuário possa escolher qual conjunto de parâmetros é mais adequado para cada caso em particular. Desta forma, os valores dos parâmetros correspondentes aos pontos não-dominados da Figura 38 são apresentados na Tabela 9.

Considerando a Figura 38, a Tabela 8 e a Tabela 9, é possível notar que o conjunto de parâmetros representado pelo ponto 3, apesar de não ter obtido uma média de ligações tão boa quanto o ponto 1, despendeu aproximadamente $1/3$ do tempo gasto por aquele. Com relação ao ponto 2, a média de ligações possui uma diferença mínima e o tempo de processamento do ponto 3 é pouco mais de $1/3$ do ponto 2. Ainda, dentre os três primeiros pontos, o ponto 3 foi aquele que encontrou o indivíduo com maior valor na coluna “Máx. Ligações” da Tabela 8.

Como o objetivo é encontrar um conjunto de parâmetros a ser pré-definido para o usuário, decidiu-se escolher um ponto em que houvesse um balanceamento entre a média de ligações encontrada e o tempo computacional requerido. Portanto, o ponto 3 foi o escolhido, sendo que os valores dos parâmetros para o operador de Otimização Parcial são os seguintes: $\textit{ProbOtimParcial} = 8\%$ e $\textit{TamOtimParcial} = 7$ resíduos.

Tabela 8: Resultados de *ProbOtimParcial* e *TamOtimParcial*.

Exp.	*ProbOtimParcial*	*TamOtimParcial*	Média Ligações	Média Gerações	Máx. Ligações	Tempo (s)
64	0	0	50,91	226,16	59(1x)	33,46
99	1	5	54,69	265,21	62(3x)	34,14
100	1	6	55,85	261,44	62(2x)	36,53
101	1	7	57,04	270,50	63(1x)	42,78
102	1	8	58,10	265,37	65(1x)	114,97
103	2	5	55,76	258,77	63(1x)	36,66
104	2	6	56,46	267,51	63(3x)	40,75
105	2	7	58,48	264,22	66(1x)	54,34
106	2	8	59,55	258,90	67(1x)	186,33
107	3	5	55,76	266,52	63(1x)	38,42
108	3	6	57,81	262,17	65(2x)	45,15
109	3	7	59,05	264,27	65(1x)	66,41
110	3	8	59,83	254,09	66(4x)	265,59
111	4	5	56,06	255,43	64(1x)	41,58
112	4	6	57,01	252,69	64(1x)	49,50
113	4	7	58,99	250,17	65(1x)	77,03
114	4	8	60,04	247,59	66(2x)	342,54
115	5	5	56,46	260,24	65(1x)	43,99
116	5	6	57,95	251,58	64(1x)	54,00
117	5	7	59,48	251,35	65(5x)	87,90
118	5	8	61,16	248,79	67(1x)	430,96
119	6	5	56,20	242,86	61(6x)	46,41
120	6	6	57,83	251,58	67(1x)	58,22
121	6	7	59,17	250,90	65(1x)	100,92
122	6	8	60,94	231,97	67(2x)	506,08
123	7	5	56,97	254,96	64(3x)	48,63
124	7	6	58,22	244,31	65(1x)	62,72
125	7	7	59,70	248,89	65(1x)	112,53
126	7	8	61,00	237,69	67(2x)	582,51
127	8	5	56,68	241,70	63(3x)	49,12
128	8	6	58,56	237,96	65(1x)	65,13
129	8	7	59,85	236,10	68(1x)	122,13
130	8	8	60,81	228,73	66(4x)	660,71

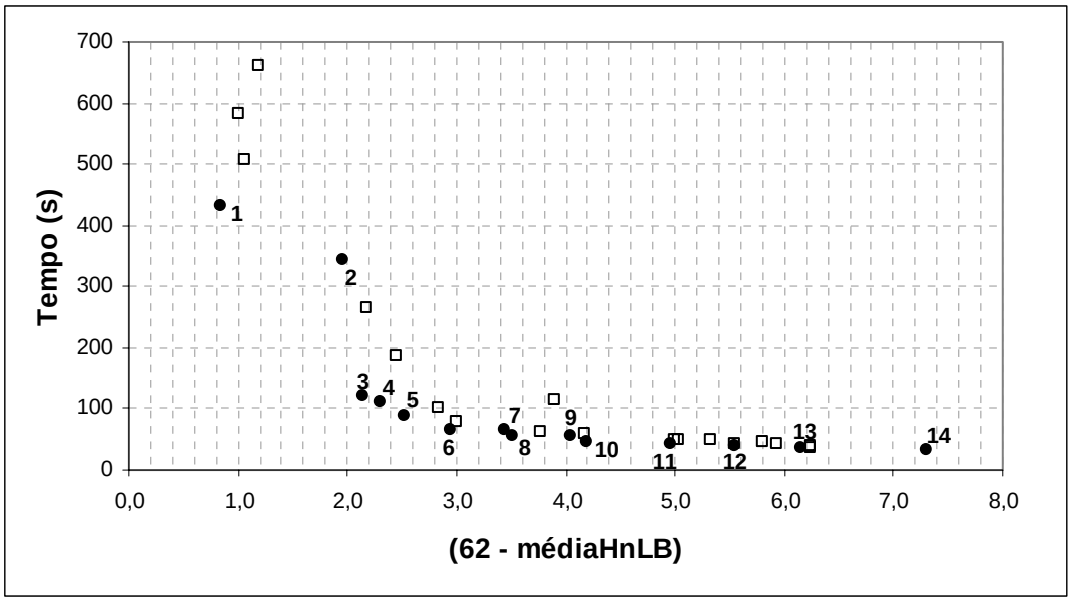


Figura 38: Gráfico de Pareto para *ProbOtimParcial* e *TamOtimParcial*.

Tabela 9: Parâmetros para os pontos não-dominados da Figura 38.

Ponto	***Prob OtimParcial***	***Tam OtimParcial***	(62 – *médiaHnLB*)	**Tempo (s)**
1	**5**	**8**	0,84	430,96
2	**4**	**8**	1,96	342,54
3	**8**	**7**	2,15	122,13
4	**7**	**7**	2,30	112,53
5	**5**	**7**	2,52	87,90
6	**3**	**7**	2,95	66,41
7	**8**	**6**	3,44	65,13
8	**2**	**7**	3,52	54,34
9	**5**	**6**	4,05	54,00
10	**3**	**6**	4,19	45,15
11	**1**	**7**	4,96	42,78
12	**2**	**6**	5,54	40,75
13	**1**	**6**	6,15	36,53
14	**1**	**5**	7,31	34,14

4.1.6 Utilizar dizimação

Para realizar estes experimentos, os parâmetros utilizados correspondem àqueles definidos nas seções anteriores, já que os parâmetros *TamTorn*, *ProbMutSempreMelhor*, *ProbOtimParcial* e *TamOtimParcial* são afetados conforme descrito na seção 3.7.1. Foram realizados 2 experimentos, a partir dos parâmetros já definidos nas seções anteriores, de modo a avaliar o comportamento do AG com e sem a utilização da estratégia de dizimação. Os resultados dos experimentos são mostrados na Tabela 10, onde se pode verificar que ao utilizar a estratégia de dizimação, o algoritmo alcança um resultado pior do que quando a dizimação não é aplicada.

Tabela 10: Resultados de “Utilizar Dizimação”.

Exp.	Dizimação	Média Ligações	Média Gerações	Máx. Ligações	Tempo (s)
131	**NÃO**	62,70	230,47	68(1x)	157,93
132	**SIM**	60,63	122,08	67(1x)	156,17

Isto pode ter acontecido devido ao fato de que quando a população é dizimada, o operador de Mutação Sempre Melhor é aplicado com o dobro de sua probabilidade normal. Como neste caso a probabilidade de Mutação Sempre Melhor é zero, ou seja, todas as mutações são simples, o algoritmo pode não ter tido condições suficientes (dadas pelos operadores) para recuperar o material genético perdido na dizimação, estagnando o processo de evolução. Portanto, foi definido que, como padrão, a estratégia de dizimação não será utilizada.

Pode ser que com uma análise mais aprofundada dos parâmetros que influenciam esta estratégia possa se extrair todo o potencial esperado dela, de forma a permitir que a evolução possa continuar após sua aplicação.

4.1.7 Dobramento progressivo

Para a realização destes experimentos foram utilizados todos os parâmetros previamente estabelecidos. Neste caso, o parâmetro “Número de Partes” referente a esta

estratégia foi variado segundo o conjunto {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 130}. Os resultados obtidos são apresentados na Tabela 11.

Considerando os valores mostrados na Tabela 11, percebe-se que à medida que o número de partes aumenta, os resultados também acompanham este aumento, com algumas pequenas variações. Mas acontece que, com o aumento do número de partes, a quantidade de resíduos referentes a cada uma diminui até um ponto em que a maioria das partes é composta de apenas 1 ou 2 resíduos, deixando a última delas com um tamanho maior (de forma a garantir que as partes possuam tamanhos inteiros, já que não se pode ter partes com quantidade fracionária de resíduos).

Tabela 11: Resultados da estratégia de dobramento progressivo.

Exp.	Num. Partes	Média Ligações	Média Gerações	Máx. Ligações	Tempo (s)
131	1	62,74	156,01	68(2x)	146,92
133	2	63,46	239,56	68(2x)	135,29
134	3	63,88	270,35	70(1x)	137,22
135	4	63,86	273,82	70(2x)	140,41
136	5	63,81	286,89	70(1x)	145,91
137	6	64,73	284,71	70(4x)	146,63
138	7	64,54	288,49	70(2x)	158,74
139	8	63,95	289,56	71(1x)	152,56
140	9	64,92	291,53	70(2x)	151,75
141	10	64,21	291,69	71(2x)	171,79
142	130	64,62	297,71	69(7x)	206,35

Conforme comentado na seção 3.7.3, quando o dobramento progressivo é aplicado, a parte inicialmente submetida ao algoritmo é a última, ou seja, a maior delas (caso a divisão do tamanho da proteína pelo número de partes não resulte em um número inteiro). Isto também acontece para o número de gerações que passarão antes que a próxima parte seja acrescentada.

Por exemplo, no experimento 142, a proteína é dividida em 129 partes de 1 resíduo, mais 1 parte composta de 11 resíduos (lembrando que a proteína tem 140 resíduos). Com relação ao número de gerações, a parte final composta por 11 resíduos será inicialmente evoluída por 42 gerações. Após este período evolutivo, serão acrescentadas as partes

compostas de 1 resíduo, sendo que após cada acréscimo, a nova proteína terá apenas 2 gerações para evoluir antes que a próxima parte seja adicionada. Com isto, percebe-se que quanto maior o número de partes, a relação entre tamanho das partes e número de gerações reservadas para que cada parte possa evoluir vai sendo cada vez mais desbalanceada.

É interessante lembrar que o operador U-*fold* é aplicado a segmentos retos existentes nas conformações, sendo que estes devem ser de, pelo menos, $1/10$ da quantidade de resíduos da proteína. Quando o tamanho das partes em que a proteína é dividida passa a ser menor que este tamanho mínimo estabelecido, o operador de U-*fold* perde sua utilidade. Assim, dividir a proteína em uma quantidade de partes menor do que 10 (cada parte sendo maior do que $1/10$ do tamanho da proteína) passa a ser uma escolha razoável.

Com base nestas considerações e considerando-se a pequena variação de valores médios entre os experimentos, foi escolhido como padrão o valor do parâmetro utilizado no experimento 140 (9 partes), já que este obteve o melhor resultado em média de ligações.

4.1.8 Considerações finais

A opção “Melhorar Última Geração” será mantida ativada, já que, no pior caso, o melhor indivíduo já encontrado pelo AG será mantido, ou seja, esta opção só tende a melhorar o resultado final e testes preliminares mostraram que o tempo gasto por esta estratégia não é significativamente maior.

A partir dos resultados apresentados e comentados nas seções anteriores, pré-determinou-se um conjunto de parâmetros passível de ser utilizado com problemas de predição de estruturas de proteína para este modelo. Isto não significa dizer que todas as instâncias serão solucionadas com estes parâmetros; eles são apenas um ponto de partida no qual o usuário pode iniciar seus experimentos. Certamente, proteínas com cadeias muito longas exigirão uma configuração mais complexa e, possivelmente, computacionalmente mais custosa. Os parâmetros estabelecidos como para o sistema “GANDALF PRED” são descritos na Tabela 12.

Tabela 12: Parâmetros para o sistema GANDALF PRED.

Parâmetro	**Valor**
TamPop	500
NumGer	300
TamTorn	3 %
ProbCross	90 %
ProbMut	2 %
ProbMutSempreMelhor	0 %
ProbUFold	10 %
ProbGeraLoops	20 %
ProbOtimParcial	8 %
TamOtimParcial	7 resíduos
Utilizar Dizimação	NÃO
Melhorar Última Geração	SIM
Dobramento Progressivo	9 partes

4.2 TESTES DO ALGORITMO

Vários experimentos foram realizados com o objetivo de estudar a influência dos parâmetros no comportamento do AG durante o processo evolutivo. Além disso, foram realizados vários testes utilizando algumas seqüências de resíduos cujo objetivo é verificar o desempenho do AG no que diz respeito a sua capacidade, como método de busca e otimização, de encontrar conformações com um número de ligações hidrofóbicas não-locais que correspondam à conformação nativa das seqüências no modelo 2D HP.

4.2.1 Seqüências de *benchmark*

O primeiro conjunto de testes realizado foi feito utilizando-se seqüências de resíduos cujo número máximo de ligações hidrofóbicas não-locais é previamente conhecido. Estas seqüências foram extraídas dos trabalhos de Unger e Moult (1993c) e König e Dandekar (1999a). A Tabela 13 apresenta a quantidade de resíduos de cada cadeia utilizada para os

testes (*NR*), a seqüência de resíduos e o número máximo de ligações hidrofóbicas não-locais conhecidas segundo o modelo 2D HP (*HnLB*).

Tabela 13: Seqüências utilizadas como *benchmark*.

NR	**Seqüência**	***HnLB***
20	HPHPPHHPHPPHPHHPPHPH	9
24	HHPPHPPHPPHPPHPPHPPHPPHH	9
25	PPHPPHHPPPPHHPPPPHHPPPPHH	8
36	PPPHHPPHHPPPPPHHHHHHHPPHHPPPPHHPPHPP	14
48	PPHPPHHPPHHPPPPPHHHHHHHHHHPPPPPPHHPPHHPPHPPHHHHH	23
50	HHPHPHPHPHHHHPHPPPHPPPHPPPPHPPPHPPPHPHHHHPHPHPHPHH	21
60	PPHHHPHHHHHHHHPPPHHHHHHHHHHPHPPPHHHHHHHHHHHHPPPPHHHHHHPHHPHP	35[7]
64	HHHHHHHHHHHHPHPHPPHHPPHHPPHPPHHPPHHPPHHPPHPPHHPPHHPPHPHPHHHHHHHH HHHH	42
85	HPHPHHHHHHHHHHHHHHHHPPPPHHHHHHHHHHHHPPPPPPHHHHHHHHHHHHPPPHHH HHHHHHHHHPPPHHHHHHHHHHHHPPPHPPHHPPHHPPHPH	52

Para cada um dos testes, o AG foi executado 100 vezes utilizando os parâmetros ilustrados na Tabela 12, exceto pelo fato de não utilizar a estratégia de dobramento progressivo, pois as seqüências são muito pequenas para que esta estratégia possa ser útil. A Tabela 14 apresenta os resultados obtidos pelo *software* GANDALF PRED juntamente com os resultados obtidos pelos trabalhos de Unger e Moult (1993c) e König e Dandekar (1999a).

É importante destacar que os resultados apresentados por Unger e Moult representam o melhor indivíduo de um total de 5 rodadas enquanto König e Dandekar rodaram a mesma quantidade que o *software* GANDALF PRED, ou seja, 100 vezes.

[7] Segundo Unger e Moult (1993c), o número máximo de ligações considerado para esta seqüência era 34.

Tabela 14: Comparação de resultados utilizando um *benchmark*.

Tam. Seqüência	Unger e Moult		König e Dandekar		GANDALF PRED	
	Máx. Ligações	Média Ligações	Máx. Ligações	Média Ligações	Máx. Ligações	Média Ligações
20	9	---	9 (100×)	9	9 (74×)	8,74
24	9	---	---	---	9 (63×)	8,61
25	8	---	---	---	8 (69×)	7,69
36	14	---	14 (8×)	12,40	14 (6×)	12,44
48	22	---	23 (1×)	18,50	23 (2×)	20,06
50	21	---	---	---	21 (6×)	18,36
60	34	---	---	---	35 (1×)	32,42
64	37	---	37 (1×)	29,30	40 (1×)	33,58
85	---	---	46 (1×)	40,80	51 (2×)	45,74

As conformações que representam os melhores indivíduos obtidos pelo *software* GANDALF PRED estão ilustradas na Figura 39, Figura 40, Figura 41, Figura 42, Figura 43, Figura 44, Figura 45, Figura 46 e Figura 47. Quando foram encontradas mais de duas conformações com a mesma quantidade de ligações, uma delas foi escolhida para ser mostrada, obviamente a que possuía o melhor valor de *fitness*. Os pontos escuros representam os resíduos hidrofóbicos enquanto os brancos são os hidrofílicos. O ponto maior indica o início da cadeia.

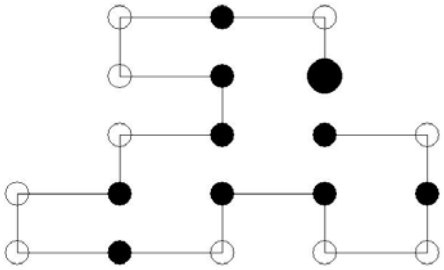


Figura 39: Melhor conformação para a seqüência de 20 resíduos.

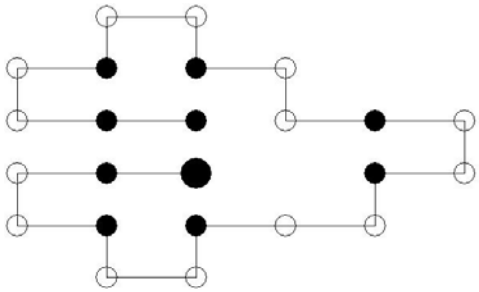


Figura 40: Melhor conformação para a seqüência de 24 resíduos.

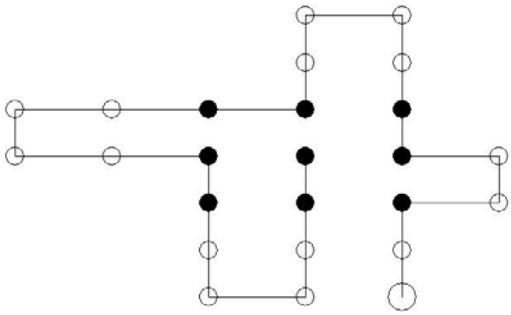


Figura 41: Melhor conformação para a seqüência de 25 resíduos.

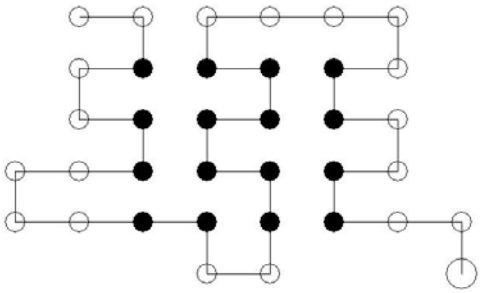


Figura 42: Melhor conformação para a seqüência de 36 resíduos.

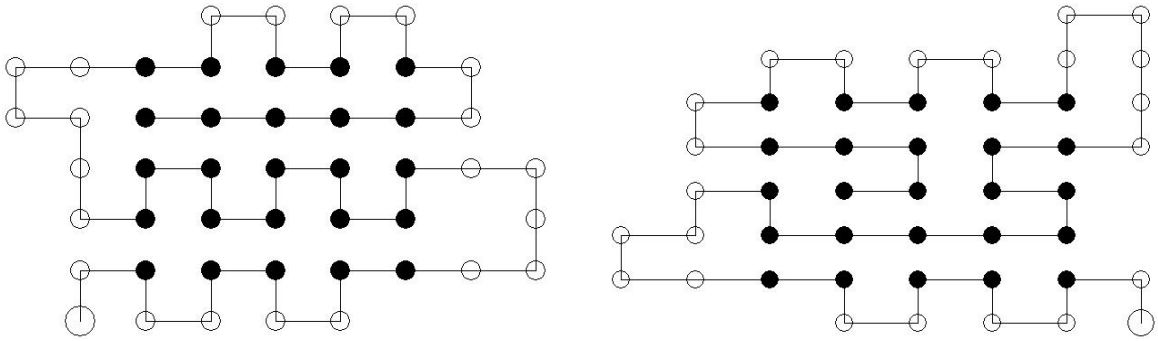


Figura 43: Melhores conformações para a seqüência de 48 resíduos.

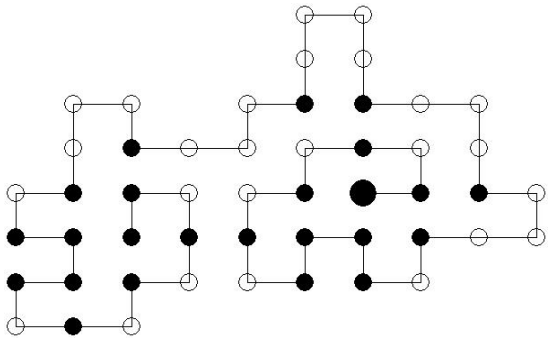


Figura 44: Melhor conformação para a seqüência de 50 resíduos.

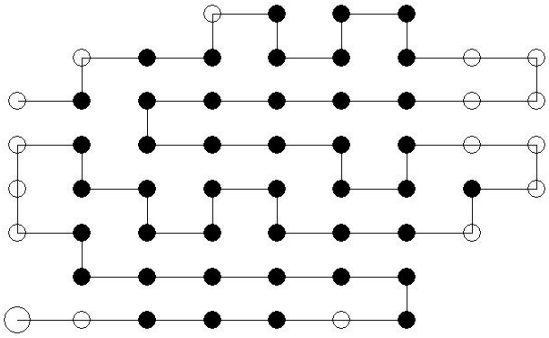


Figura 45: Melhor conformação para a seqüência de 60 resíduos.

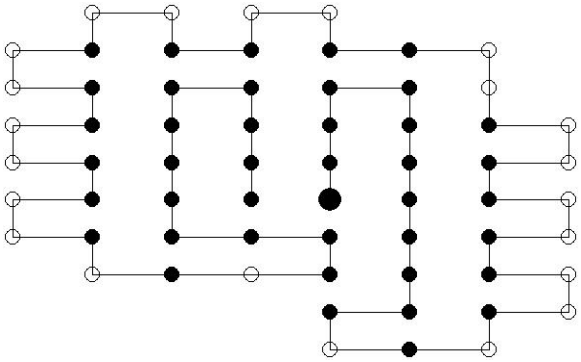


Figura 46: Melhor conformação para a seqüência de 64 resíduos.

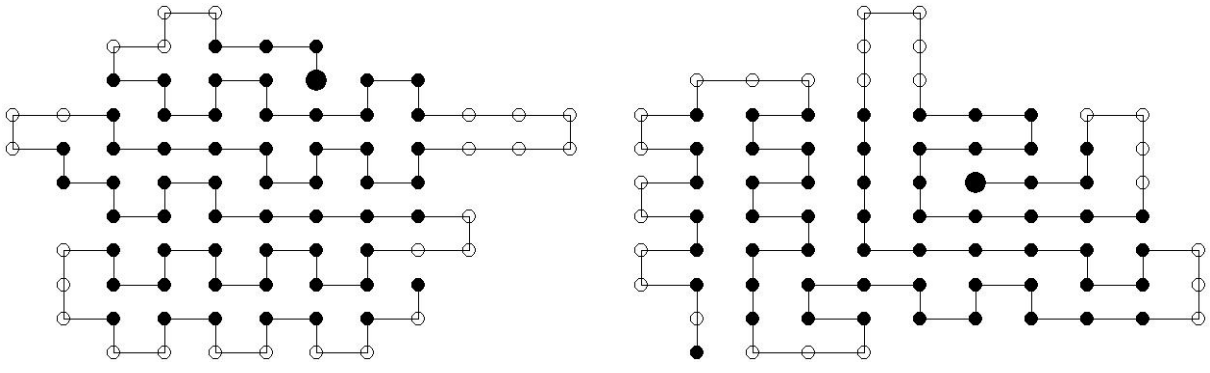


Figura 47: Melhores conformações para a seqüência de 85 resíduos.

4.2.2 Seqüências de proteínas reais

Um segundo conjunto de testes realizados com o *software* GANDALF PRED utilizou algumas proteínas selecionadas do PDB. O PDB é uma referência na área de proteínas sendo que todas as proteínas contidas nele possuem sua estrutura terciária precisamente determinada por meio de cristalografia de raio-X ou ressonância magnética nuclear. Os dados extraídos do PDB são considerados confiáveis e as conformações das proteínas representam seu estado nativo real.

O arquivo do PDB referente a uma determinada proteína possui informações detalhadas a respeito de sua estrutura. Porém, como o algoritmo implementado utiliza o modelo HP, é necessário "traduzir" a seqüência de resíduos da proteína, que utiliza os 20 aminoácidos padrão, para uma seqüência de Hs e Ps para que possa se encaixar ao modelo utilizado. Contudo, nem todos os aminoácidos são totalmente hidrofóbicos ou totalmente polares. Assim, uma matriz de tradução, baseada nas características químicas dos aminoácidos, é usada para definir se um aminoácido será considerado hidrofóbico ou polar. Porém, existe certa divergência entre o PDB e a literatura, com relação a quais aminoácidos são definidos como hidrofóbicos e hidrofílicos. No entanto, decidiu-se utilizar a definição utilizada pelo PDB no que diz respeito à matriz de tradução. A matriz utilizada está apresentada na Tabela 15, onde as colunas representam os aminoácidos e as linhas indicam como um determinado aminoácido é considerado.

Tabela 15: Matriz de tradução.

	A	**R**	**N**	**D**	**C**	**Q**	**E**	**G**	**H**	**I**	**L**	**K**	**M**	**F**	**P**	**S**	**T**	**W**	**Y**	**V**
H	×				×			×	×	×	×	×	×	×			×	×	×	×
P		×	×	×		×	×								×	×				

Para a realização destes testes, foram utilizadas 7 proteínas. Estas proteínas foram escolhidas por possuírem características globulares, ou seja, suas conformações tridimensionais são bem compactas. Esta escolha foi feita devido ao fato de que a função de *fitness* tenta refletir a atração entre aminoácidos hidrofóbicos e sua concentração no interior da conformação. A Tabela 16 apresenta as 7 proteínas escolhidas traduzidas para o modelo HP juntamente com seus códigos de identificação no PDB (ID) e seu respectivo o número de aminoácidos (NAA).

Os testes foram realizados com os parâmetros mostrados na Tabela 17, pois testes preliminares indicaram que o algoritmo conseguia obter melhores resultados com estes parâmetros em comparação com aqueles definidos na Tabela 12. Também foi utilizada estratégia de dobramento progressivo, sendo que o número de partes para cada proteína foi configurado para que as partes tivessem, em média, 5 resíduos cada uma, ou seja, o número de partes utilizado para cada proteína, da menor para a maior, foi {50, 60, 75, 75, 100, 150, 150}.

Tabela 16: Seqüências de proteínas reais.

ID	NAA	Seqüência de resíduos
1d9i	288	HPHHHPPHHPHHPHPPHHPPPHHPPHHPHPHHPHPPHPHHHPPHPHHHHPHPPPPHHHHH PHHPPPHHHHHHHHHHHPPHPHPHHPPPHHHPHHHHPPHPHPPPHPHHPHHPHHHHPPH PHPPPHPPPHHHHHHHHPHHHPPHHHPHHHPPPPHHHPHHPHHHHHPHHHHHPHHPHHHH PHHHHPPPHHPHHPHPHHPPPHHHPPHPHPHHPPHHHHHHHPPPHHPHPHHPHPPHHPHH HHPPHPPPHHPHHHHPHHPHHPPPHHHHHHHHHHPHHHHHPHHHPPHH
1epr	330	PHHPHHHHPHPPHPPHHHHPHPHHHPHPHHPHPHPHHPPPHHHHPPPHHHPPHPHPHHHH PPHPHHHHHHPHHHHPHPHHPHPPPPHPHHHPHHPHHHHHHHHHPHHPPHHHHPPPHHPPP HHPHHHHHHHPHHPHHPPHPPHHHHPPHHHPHPPPHHHHPHHHHHPHHHPHHHHPHHHHH HPHHHHHHPHHPHHHPHHPHHHHHHPHHHHPHPHPHHHPHHHHHHHHPHHHHPHHHHPHP HHHPPPPHHHHHHPHPHHHPPHHHHHHPHPHHHPHPHHPHPHPHHPPPHHHHHPPPHHH HHPHHHPHHHHHHHHHHPHHHHPHHHHHPH
1axk	394	HPHHPHPPHPHHHHHHHPHPHHPHHPHHHHPPHHHHHHPHHHHPHPPHPHPHHHHPHHH PHPHHHPHHHHHPHHHPHHHPHPHHHHHHHHHPHPPHHHHHHHHPHHHHHHHHHPHPPHPHH HHHPHHPHHHHPPHHHPHPHHPPHHHPHPHHHHHPPHPHPHHPPHHPHHHHHPHHPHPHH PHPHPHPPHHPHHHHHHHHHPPHPHHPHPHHHHHPPHPHHHHHHHHPPPHHPHHHHPP HHHHPPHHHHHHHHHPPHHHHPHHHHHPHPHPPHPHPPHHHHPHHPHPPPHPPHHPPHHH HHPPHHPHHHPHHHPHPPHHHPHHHHPHHPPPHPPPHHHHHPHHHPPHPHHPPPHHPHH PHHPPHHHHPHHHPHPPHPHHPPHHHHHHHHPPH
1deq	411	HHHHPPPHHHHPPPHHPHHPHHHHHHPHHPPHPHPHPHPHHPHHHHPHPPHHPPHPPH HHHHPHPHPPPPPPHPPPHPPHHHPPHPHHPPHHHHPHHHPHHPPHHPHHPPHHPPPPPH HHPHPPHHHPHPHPHPPPHPPHHHHHPHHHPPHPPHHPHHHHPPHHHHHPPHHHHPPHHH HHPHPHPHPHHHHHPHPHPHPHHHPHHPHHPHHHHHHPPHHPHPHHHHPPHHHHHPHPP HHPHHHPHPHPPHPHPHPHHPHHPHHHHHPPPHHPHHHHHHHHHPHHPHPHHPHPHPPPP PHHHHPHPHHPHPHHPPPPPHHPHPHHPPPHPHHHHPHHHHHHHPHHHHPHHHHPHHPHP PHHPPHHHHHHHHPPHHPHHHHHHHHHPHPPHHHHPHPPHPHHHHHPHHPH

Tabela 16: Seqüências de proteínas reais. (continuação)

ID	NAA	Seqüência de resíduos
1mr5	648	HHHPHHHHHHHHHPHHPHPPPHPHHHPPPPHHPHPHPHHHHPPHHHPHPHPHHHPHPHHH HHHHPHPHPHPHPPPHHPHHHHHPHPPHPHHPHPHHHHPPPHPPHPPHHPPHHHHHHPHH HHHHHPHPPPPPHHHPHHPHPPHPHHHHHHPHHHPHHHHHHHHPHHHHPPHPHHPHHPHP HPHHHHPPHHHHHHHHHHHPPHPHHHPHPHHPHHHPHHPHHHHPPPPHHHHHHHHHPPHH HHPPPHHHPHPHHHHHPHPHPPPHHHPPPPHHPHHHPHHHHHPPHHHPPPHPPPPHPP PPHHHHHHPHHPHHHHHHPHPHHHPHHPPPPHPHHHHPPPPHHPHHPHPHHPPPPHHPPHH HHPPHHHHHHPHPPPPHHPHHHHPHHHPHPHHHPHHPPHHPHPPHHPPHHHPHPPHHPPP PPHHHPHHHHHHHHHHHPHPHHHHPHPPHHPHHPHPHHPHPPHPPHHHHHHHHHHHHPH PPPHPPPPHHHHPHHHHHHHPHHHHPHPHHHPPHHHHPHPPPHHHHHHHPHPHPHPPP HHHPHPHHPHHPHPHHHPHHHHHHHHPHHHPHHHPHPPHPHPHPHHHPPPPHPPHPHHHH HHHHPPHHPHPPPHHHPPHHHHPPPHPHPPHPHHHHPPPHHHPHHPHHHHPHHHP
1fgh	753	PHHHHHPHHPPHPHHPHPHHPHPHPHHPHPHPPPHHHPPHHHHHHHHPPPHPPPHPPHHHH HPHPPPPHHHPPHHHPHHHHPHHPPHHPHHHHPPHHHHPHHHPHPHHHPHPHPPHHPHPP PHHPHHHHHHHHHHHHHHPPHPHHHHPHHHPPHHHPHHHHHHHPPHHPPHHHHHHHHHH HHHPHHPHHHHHPHPHHHPHHHHHHHHHPHPHHHPPHPHHHHHHHHHHHHHHHHPHH HPHHPPHPHHHHHHHHPHHHPHHHHHPHHPHPHHHHHPHHHPHPHHPHHPPHHPHHHP PPHHHHPPHHPHPHPPHHPHHPHPHHPPHHHPHHPHHPHHPHPHHPHPHPHHHHHPHPP PHPPHHPPHHHHHPHHHHHHHHHPPHHHHPHPPPHPHHHPPPHHHPHHPPHHHHHHPHH HPHHHPHPPHPHHHHPHPHHHHPHPPPHHHPPPHPPPHHHHHHPPPHHHHHHHHHHHHP PPHPHHHHHPHHHHHHPHPPHPPHPPHPHPPHPPHHPHPPHPPPHPPHPHPPHPPPHPHH PPHPHHPHHPHPPHPHHHHHHHHHHPHHPHHHPHHHHPHHHPPHPPPHHHHHHPPPPPH HPPHPPHHHPPHHPHPPHHPHHHPHHHPHHHHHPPPHHPHPPPPHPHHPPPHHHPHHHH HPHHPHHPHPHHHPHHHPHHHHPPHPHPHHHPHPHHHHPHHHPHHPHPHHHHHHHPPHH PPHHHHPHHHPPHPHPHHPHHPHHPPHHPHPPH
8gpb	842	PPPHPPPPHPHPHPHHHHHPPHHPHHHPHPPHHHHHHHHPPPHHPPPHHHHHHHHHP PHHHHPHHPHPPHHHPHPPHPHHHHPHPHHHHPHHPPHHHPHHHPPHHPPHHHPHHHPHP PHPPHPPPHHHHPHHHHPHHHHHHPPHHHHHHHHHHHHHHPHPHHHHPPHHHHHHPHPPHP PHHPHHPPHPHHPPPHHHPHHHHHPHPHHPPHHHHHPHPHHHHPHPHPHHPPPHHPHH HPHHPHHHPPPHPHHPHPHHHHHPHHHPPPHHPPHPPHHHPPPPHHPHHPHPHHPPHHHH HHHHPPHHPPHHPPHHHHPPPHPHPHHPPHHHHPHPPHHPPHHHPHHPPHHPHHHHPPPHP HPHHHPHHHHHHHHHPHHHHPPHHPPHPHHHHPHHHPPHHPHHHPHPPPHHPPHHHHHPH PHPPHPPHPHHPPHHHHPHPHHHHHHHHPHHHPHHHPHHPPHHHHHHHHPHHPHPPHHHP PHHPHHHPPPHHHHHPPHHHPHHHPPHHPPHHPPHPPHPHHHPHHPPPHHHPPHHHHHPP PHHHHHHHHPPPHHHHHPPPPHHPHPHHPHHPHHPPHHPHHHHHHHHPPHHHPPPHHHHP PHHHHHHHHHPHHHHHHHHHHHHHHHHPHHPHPPHHHPPHPHHHHPPHPHPHHPHHHPH PHPPPHPHHHHPHPHHHPHHHHHPHHHHHHHHHPHHPHPHHPPHHPPPHHHHHHPHPPHPP HPPPHHPHPPHHPPHPPHPPHHPPHPPHHHPPHPPPHHPHHPHHHHHPPHHHHHPHPPH HHHPPPHPHHHHPPPPHHPHHHPPHHHPHHHPPPPHHHPHHPPHHHHPPPPPPHPHPPPHHP

Tabela 17: Parâmetros utilizados com as proteínas reais.

Parâmetro	**Valor**
TamPop	500
NumGer	300
TamTorn	3 %
ProbCross	90 %
ProbMut	2 %
ProbMutSempreMelhor	30 %
ProbUFold	0 %
ProbGeraLoops	0 %
ProbOtimParcial	8 %
TamOtimParcial	7 resíduos
Utilizar Dizimação	NÃO
Melhorar Última Geração	SIM

É importante ressaltar que o número máximo de ligações hidrofóbicas não-locais para estas proteínas para o modelo utilizado não é conhecido *a priori*. Assim, não é possível avaliar precisamente quão boas são as conformações encontradas, sendo que provavelmente as conformações obtidas não representam o ótimo global para estas proteínas. Deste modo, a avaliação será realizada apenas pela análise visual das melhores conformações encontradas pelo algoritmo.

Os resultados obtidos pelo algoritmo implementado estão sumarizados na Tabela 18. Para cada proteína são mostrados o número máximo de ligações hidrofóbicas não-locais (MaxHH) e a seqüência de dobramento.

A representação gráfica das conformações mostradas na Tabela 18 são apresentadas na Figura 48, Figura 49, Figura 50, Figura 51, Figura 52, Figura 53 e na Figura 54. Vale lembrar que os pontos escuros representam os resíduos hidrofóbicos enquanto os brancos são os hidrofílicos, e o ponto maior indica o início da cadeia.

Tabela 18: Seqüência de dobramento para as proteínas reais.

ID	MaxHH	Seqüência de dobramento
1d9i	126	FFFFFFFFFFFFEFFFFFFFFFFFFFFEFEEDDEEDFFFFFFFFFFDFFFFFFFFFFFED DEFDFFDDFEFFFFFFFFFFFFFEFFFFFFFFEEFFFFFFFDFFFFFFFFDDFFFFFFFEE FDFFEEFFDFFFFFEFDDFEFDDFFEDDEFFFEFFDFFFEFFDEFFEFEEDDEFFDFDDE FFFFFEEFFFFDEFDEFFDFDDEFEEDEDFFEDDEDDFFEDFEDFFEFFDEEDEEFFFDD FFFEDFEEDFEEFDEDDEEDDEDDFFFEEFFFFDDFFFFEEFFDEED
1epr	164	FFFFFDDFFFFFDEEDFFEEFFFFFFFFFFFFEFDDFEDDFFFFFFFFFFFFFFEEFFFFFF FFFFFFFFFDFFFFFDEFFFFEFDDFEDDFFFFFFFDEEDDEFEEDEFFFFFFFFFFDDF FEDDEFFFFFEDFEFFFFFFFFFFFFEEFFFFFFFFDDFFFFFFFFEFEEDFFFFFFFFEF DDFEDDFFFFFFFFFEEFFFFFFFFFDFDEFFEDDEFFDDFFFFFFFEEFFFFFFFFDED DEEDFFDEDDFDEEDFFEEFFFFFDDFFFFFEFFFFFEFFFFFFDEFEEDFFFDEEFEFF FDDFFFEEFFFDDFFFEEFFFDDFEDFDD
1axk	193	FFFFFFEEFFFFFFFFFFFFFFFEDDEFFFDDFFFFFFFFFFFFFFFFFFFFEEFFFFFF FFFFFFFFFFFFDDFFFFFFFFFFFFFFFFFEEDEEDFFFFFFFFFFFFDDFFFFFFFFF FEDFEEFDEEFDFFFFFFFFFFFFEFDDEDDEDFFFFFFFFFFFFEEFFFFFFFFFFFFDED DFDEFFFFFFFFFFFFFFDEFEEDEFFFFFFFFDEDDFDEEDDEFFFFFEFDEEDFDDEE DFEDEEFEDDEEDDEEFFFFFDDFFFFFFFFFEEFFFFFFFFFDFDDEFFFFEDDEFEEDEF FEDDEEDDEDDFFFFEEFFFFDDFEFEEDDFDFEFDDFFFFEEFFFDDFEEFDDFFDEED DEEFEFFFFDFEEDFDDEDFEDDFFEFFEEDEE
1deq	171	FFFDFFFFDFFFFFFFFFFFFFEFDDFEDDFFFFFFFFFFFFFFEEFFFFFFFFFFFF FFFFFEFFFDDFFFFDFFFFFFFFFFFFFFFFFFFFEEFFFFFFFFFFFFFDDFFFEEFFF FFFEDDEEDDFDFFFFFFFFFFFFDFEEFDFEEFFFFFFFFFFFFFDDFFFFFFFFFFF FEEFFFFFFFFFFFFFFEFDFDDEEDDFFFFFFFFFFFFFFEEDEEDFFFFFFFFFFDDF EDDEEDDEFFFFEEDEEDDEFFFFFEDDEDDFFFFFFFFFEEFFFFFDDFFFFFFDEEFE FFFFFFFFEFDFDDEEDDFFFFFFFFFEEFFFFFFFDDFFFFFFEEFFFFFFFFDFFFFD FFEDDEFEDDEFFDDEFFDDFFEFFFFFFFEFFFEFEEDDFFFFFDFDFFDDF
1mr5	265	FFFFFFFFFFFFFFFFFFFFFFDDFFFFFFFFFFFFFFFFFFFFFFFEFFFFFFFFFFFF FFFFFFFFFFEFFEEFDFFFFFFFFFFFFFFFFFFFFFDFFFFFFFFFFFFFFFFFFF FEDDEFFFFDDFFFFFFFFFFFFFFFFFFFFFFFFFFFFEEFFFFFFFFFFFFFFFFFF FFFFDFDDEFFFFFFFFFFFFFFFFFFFFFFFFFEEFFFFFFFFFFFFFFFFFFFFDDFF FFFFFFFFFFFFFFFFFFFFEEFFFFFFFFFFFFFFFFFFFFFFFDDFFFFFFFFFFFFF FFFEEDEEDDEEDFFFFFFFFFFFFFFFDFFFFFFFFFFFFEDDEEDDEFFDDFFFFFF FFFFFFFFFEFFFFFFFFFFFFEEFFFFFFFFFFFDFFFFFFFFFFFFFFDFDDEFFFF FFFFFFFEFFFFFFFFFFFDEFFEFEEDDEFFFFFFFFFFFDDFFFFFFFFFFFFFFF FFEFFEEFDFFFFFFFFFFFFFFFDDFFFFFFFFFFFFFFEEFFFFFFFFFFFFDDF FFFFFFFFFFFEEFFFFFFFFFFFDDFFFFFFFFFEEFFFFFFFFFFFDFFDDFEFFFF FFFFFFEEFFFFFFFFFFDDFFFFFFFEEDFEFFFFEEFFFDDFFF

Tabela 18: Seqüência de dobramento para as proteínas reais. (continuação)

ID	MaxHH	Seqüência de dobramento
1fgh	329	FFFFFFFFFFFFFFFFFDDFFFFFFFFFFFFFFFFFFFFFFFFFFFFEEFFFFFF FFFFFFFFFFFFFFFFFFFFFDDFFFFFFFFFFFFFFFFFFFFFFFFFFFFDEEF EFFFFFFFFFFFFFFFFFFFFFFFFFFFFDDFFFFFFFFFFFFFFFFFFFFFFFF FFEEFFFFFFFFFFFFFFFFFFFFFFFFFFFFDDFFFEDDEFFFFFFFFFFFFFFF FFFFFFFFFEEFFFFFFFFFFFFFFFFFFFFFFFDDFFFFFFFFFFFFFFFFFFF FFFEEFFFFFFFFFFFFFFFFFFFFFFFFDDFFFFFFFFFFFFFFFFFFFFEEDFE EFDFFFFFFFFFFFFFFFFFDDFFFFFFFFFFFFFFFFFEEFFFFFFFFFFFFF FFFFFFDDFFFFFFFFFFFFFFFFFFFFFDFEEFDEEFFFFFFFFFFFFFFFFF FFFFFFFDFDDEEDDEFFFFFFEDDEFFFFFFFFFFFFFFEDFDEEDFFEEFFFFFED FFFFFFFFFFFDDFEEFFDDFFFDFEFFFFFEFEEDFFFDDFFEEFFFFFFFFFFDFF EDDEFFFEFDDFEFDFDDEFFFFFFFEFFFFFFFFFFFFFEEFFFFFFFFFFFDFFFF FFDFFFFFFDDFFFFFEFFFFEFFFFFFFFEEFFFFFFFDDFFFFFFFEEFFFFFF DDFFFFFFFEFFFFEFFEFEEDDFFDDEFEE
8gpb	346	FFFFFFFFFFFFFFFFFFFFFFFFFFDDFFFFFFFFFFFFFFFFFFEEFFFFFF FFFFFFFFFFFFFFFFFEDDEFDDFFFFFFFFFFFFFFFFFFFFFFFFFFFFDE EDEEDEEDFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFDDFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFEEFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFDDFFFF FFFFFFFFFFFFFFFFFFFFFFFFFFEEFFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFDDFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEEFFFFFFFFFFFF FFFFFFFFFFFFFFFFFDDFEFDDFEFFFFFFFFFFFFFFFFFFFFFFFFFF FEEFFFFDEEDFFFFFFFFFFFFFFFFFFFFFDDFFFFFFFFFFFFFFFFFEFDDFE EFFEDFFFFFFFFFDEEDFFFEFDFEDFEEFFEDFEFDFFFFFFFFFFFFFDEFDFFFF FFFFFFFFFFFFFFEFDDFEFFDFEDFFDDFEDFEFFFFFFFFFFFFFFFFFFFEEFF FFFFFFFFFFFFFFFFFDDFFFFFFFFFFFFFFFFFFFEFFFFFFEFEEDDEEDFFD FFFFFFFFFFFFFFFFFFFFDDFFFFFFFFFFFFFFFFFFFEEFFFFFFFFFFFFF FFFFDDFFFFFFFFFFFFFFFEEDEEDFDFFDEEFEFFFDDFFFFDFEEFDDFEEFDDE DDEFEEDEFFEDEFFFFFFFFFDDFFFFFFFEEFFFFFDDEDDEEFDEFFEEFDEDFFFEF

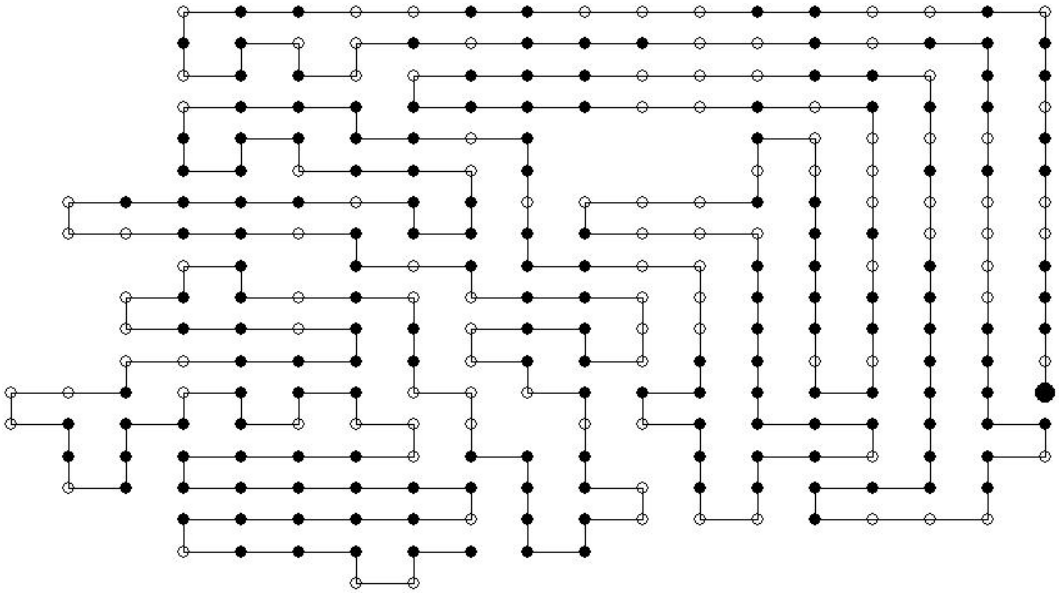


Figura 48: Melhor conformação para a seqüência 1d9i.

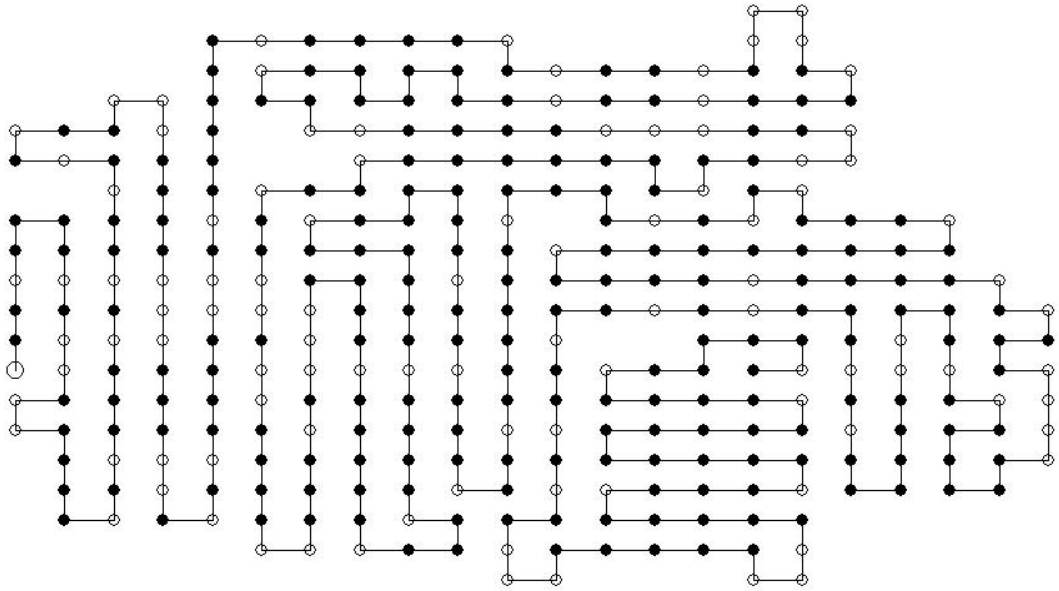


Figura 49: Melhor conformação para a seqüência 1epr.

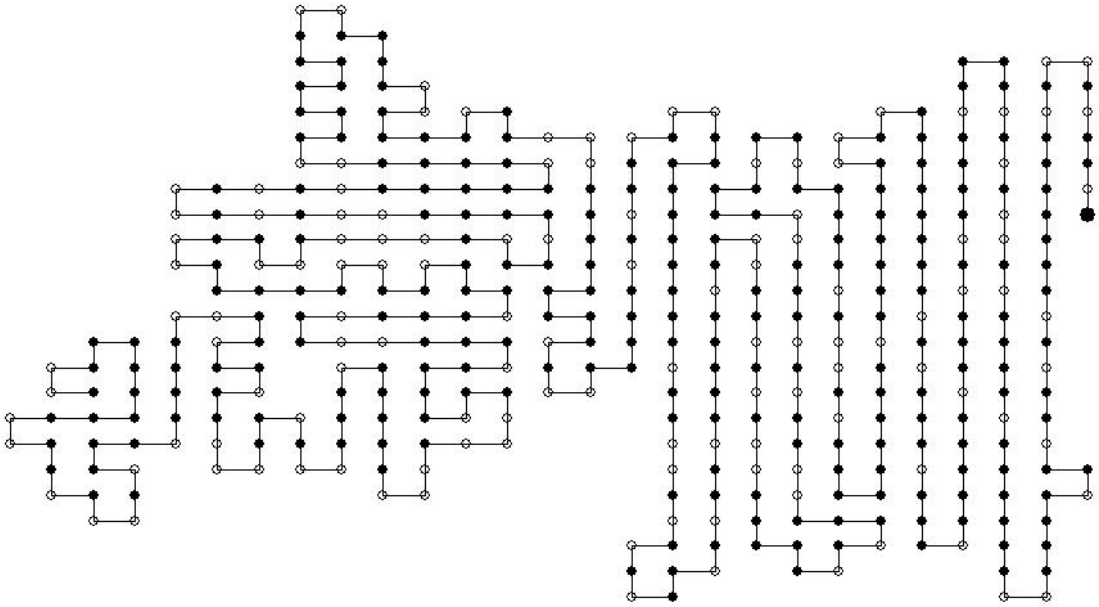


Figura 50: Melhor conformação para a seqüência 1axk.

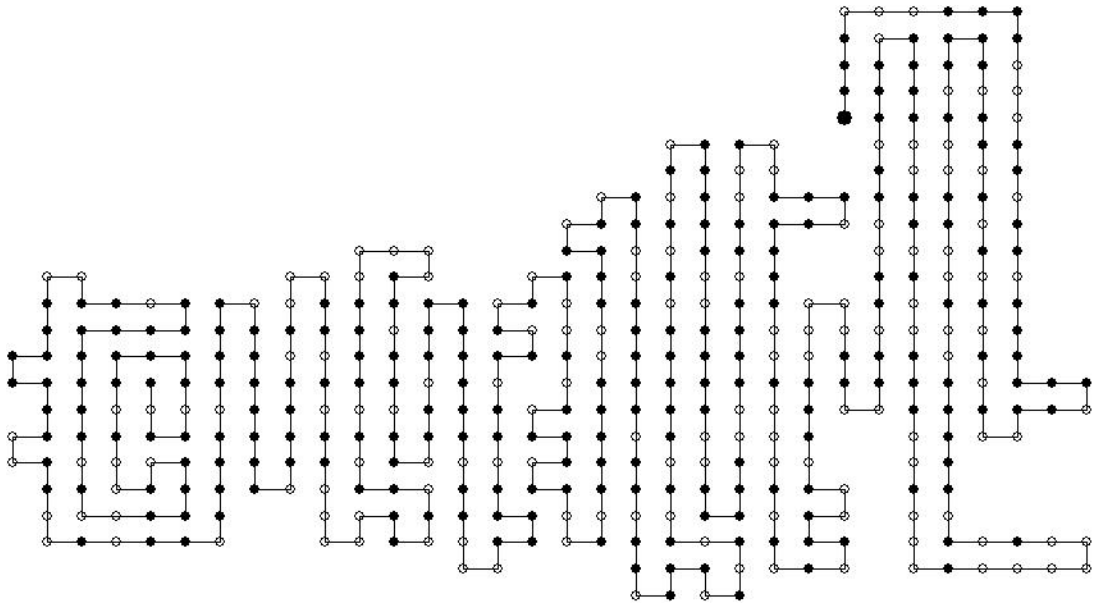


Figura 51: Melhor conformação para a seqüência 1deq.

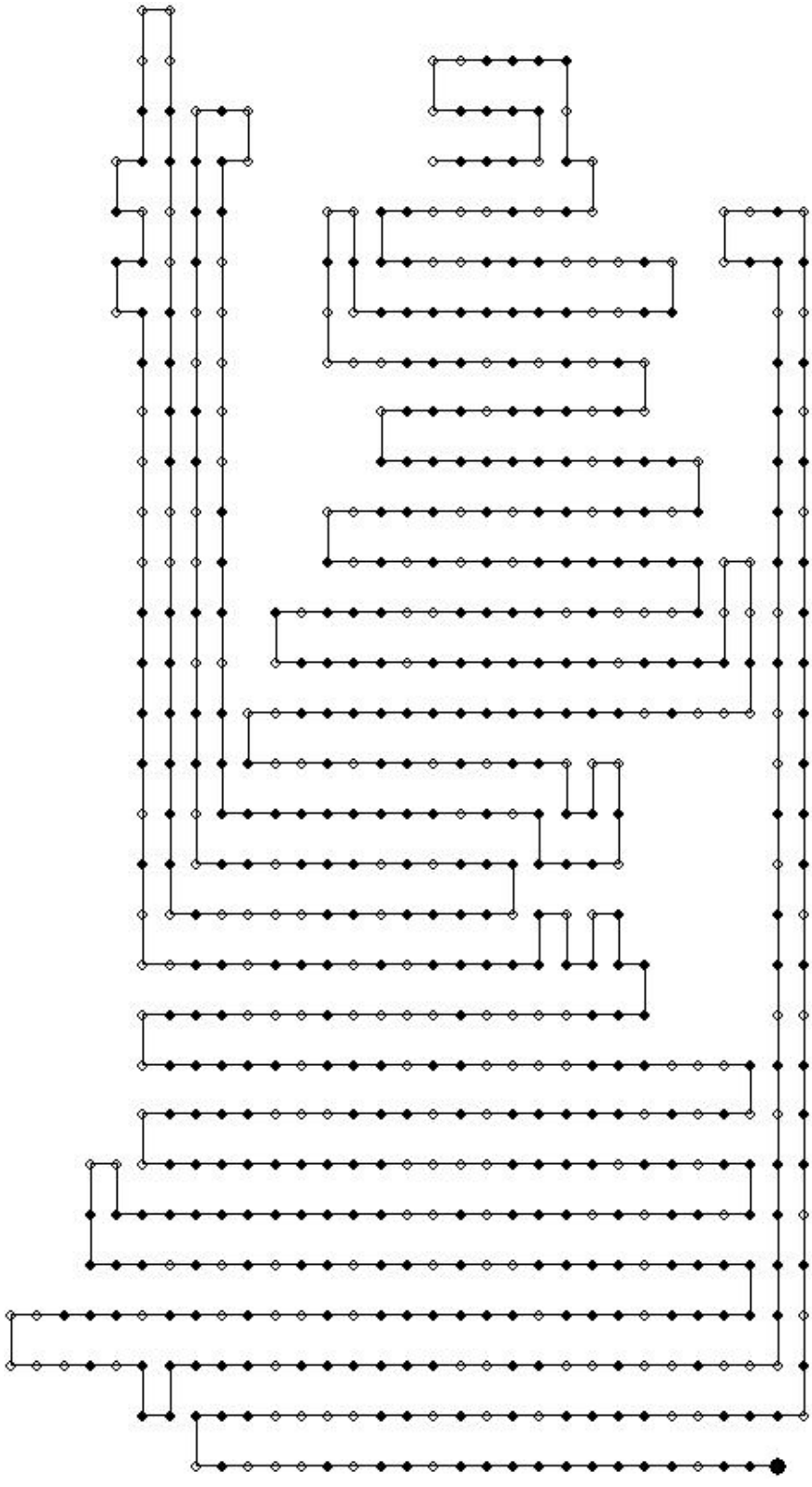


Figura 52: Melhor conformação para a seqüência 1mr5.

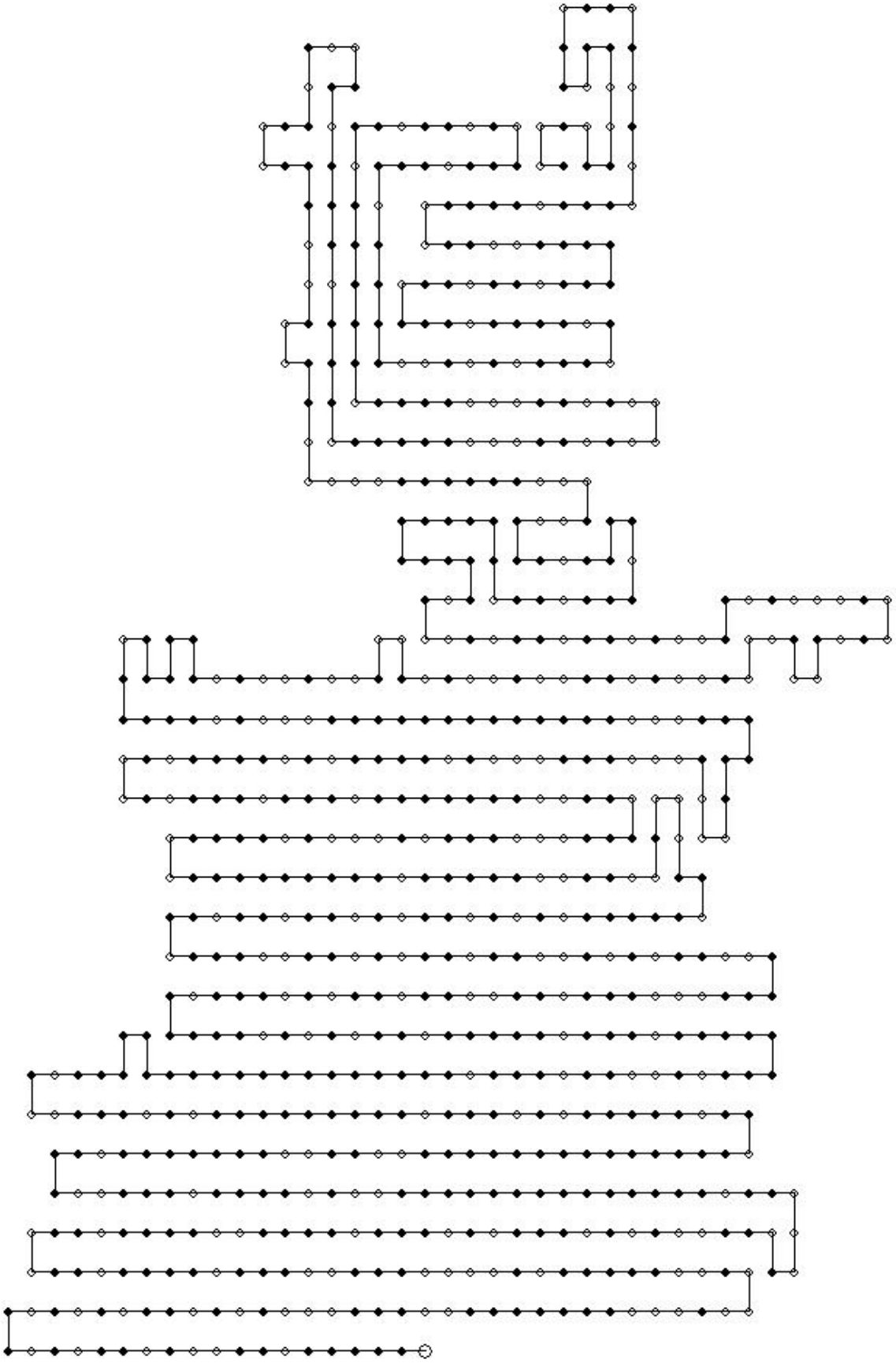


Figura 53: Melhor conformação para a seqüência 1fgh.

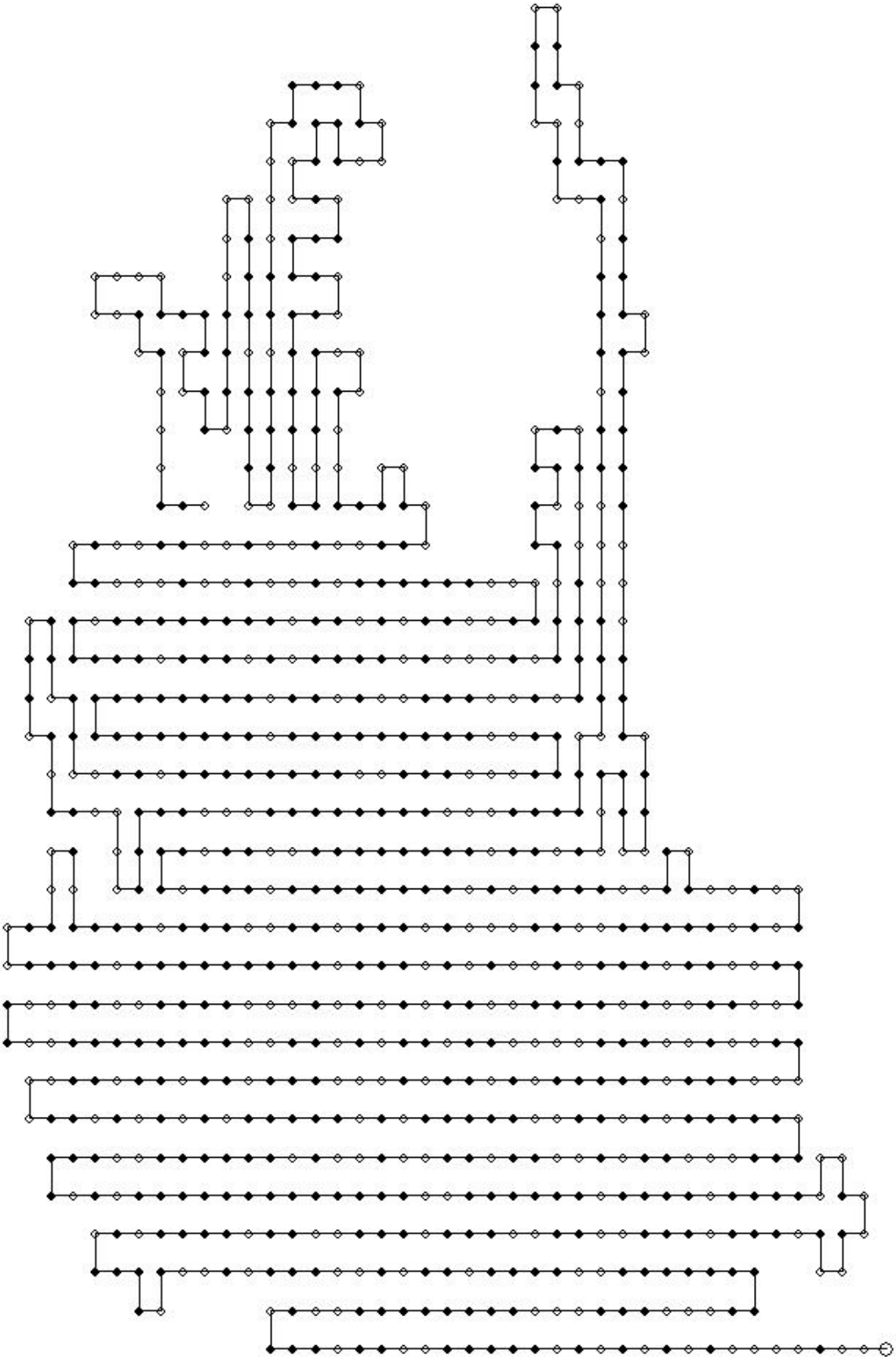


Figura 54: Melhor conformação para a seqüência 8gpb.

CAPÍTULO 5

DISCUSSÃO E CONCLUSÃO

5.1 DISCUSSÃO DOS RESULTADOS

5.1.1 Definição de parâmetros do AG

Primeiramente, foi realizada uma bateria de testes com o objetivo de estabelecer um conjunto de parâmetros que possa inicialmente ser utilizado pelo algoritmo. Estes parâmetros, provavelmente, não serão ótimos para quaisquer instâncias submetidas ao algoritmo, mas serve como uma referência inicial para que o usuário possa começar os seus testes.

Analisando a Tabela 4, que apresenta os resultados obtidos pelos experimentos para os parâmetros *TamPop*, *NumGer*, *TamTorn*, *ProbCross* e *ProbMut*, percebe-se que o conjunto de valores referente ao experimento 64 apresenta o melhor resultado em média de ligações. Porém, a variação desta média para os experimentos 52 e 61 é muito pequena. Além disto, tanto o máximo de ligações quanto o tempo para estes dois experimentos pouco variam em relação ao primeiro. Desta forma, decidiu-se por optar pelo conjunto de parâmetros do experimento 64 como padrão, baseando-se no valor da média de ligações, já que esta representa um conjunto de 100 rodadas.

Após a escolha dos valores para os parâmetros básicos do AG, a Tabela 5 mostra os resultados dos experimentos para testar a utilização do operador Mutação Sempre Melhor. Analisando estes resultados, nota-se algo interessante. À medida que a probabilidade de aplicação deste operador é aumentada, a média de ligações diminui. Este comportamento era esperado somente para valores de probabilidade relativamente altos, devido ao fato de que com muita mutação deste tipo, o algoritmo seja levado a um máximo local muito rapidamente causando perda de diversidade genética e ocasionando uma convergência prematura, conforme verificado pelo experimento 78.

Porém, o AG apresentou resultados realmente inesperados, considerando valores baixos desta mutação, chegando ao ponto em que a utilização deste operador com qualquer valor de probabilidade tenha resultado em médias de ligações piores quando comparado à não

utilização do mesmo. Isto pode ter acontecido devido à aplicação individual deste operador, ou seja, pode ser que sua utilização em conjunto com algum outro operador possa vir a apresentar melhores resultados.

Com relação ao operador *U-Fold*, cujos resultados foram apresentados na Tabela 6, nota-se, através da análise dos resultados, que a utilização deste sempre resulta em um valor de média de ligações melhor em comparação a sua não utilização, independentemente do valor de probabilidade escolhido. Isto sugere que o operador deva ser utilizado. É interessante observar que, neste caso, não foi escolhido o valor de probabilidade referente ao experimento que obteve a melhor média de ligações. A análise realizada levou em consideração todos os valores das colunas apresentadas na referida tabela, escolhendo-se o valor referente ao experimento 79. Neste experimento, a média de ligações é apenas 0,35% menor que o melhor experimento. A média de gerações foi a maior dentre todos, significando que o algoritmo conseguiu manter a diversidade genética da população por mais tempo. Para o máximo de ligações, bem como para o tempo, os valores não variaram muito. O experimento 79 obteve, juntamente com o experimento 83, o melhor valor para o máximo de ligações, perdendo apenas na quantidade de vezes em que este valor foi encontrado. Por fim, o experimento em questão apresentou o maior valor para a quantidade média de gerações que o operador foi aplicado. Porém, como o operador foi aplicado, em média, somente na primeira geração, seu efeito foi diluído ao longo das gerações restantes. Isto significa que sua função é somente aprimorar as conformações geradas no início da busca.

Para o operador Gera *loops*, cujos resultados estão apresentados na Tabela 7, também se observa que quando este operador é utilizado, o AG alcança valores melhores de média de ligações, em comparação com a sua não utilização. Porém, a diferença entre o melhor e o pior resultado é apenas de aproximadamente 2%. Desta forma, como não houve nenhuma diferença significativa entre os resultados obtidos pelos experimentos realizados para testar este operador, decidiu-se optar pelo valor de probabilidade representado pelo experimento 90, como sendo um valor em que houvesse um balanceamento entre os valores das colunas da referida tabela.

Conforme comentado na seção 4.1.5, a análise dos resultados obtidos com os testes nos parâmetros do operador de Otimização Parcial foi feita utilizando-se o conceito de Otimalidade de Pareto. Este conceito torna-se necessário quando é feita uma análise multi-objetivos. Através da Tabela 8, nota-se que o custo computacional da aplicação deste operador cresce à medida que o parâmetro *TamOtimParcial* é aumentado. Portanto, achou-se conveniente realizar este tipo de estudo para possibilitar que o usuário possa realizar a escolha

do melhor conjunto de parâmetros baseando-se em qual dos objetivos deseja-se dar mais ênfase (tempo de processamento ou desempenho na predição). No entanto, como o objetivo era selecionar um certo conjunto de parâmetros, foi realizado um balanceamento entre os dois objetivos considerados de modo a não impactar nem o tempo de processamento nem a média de ligações.

A Figura 38 mostra a representação gráfica da relação entre a média de ligações e o tempo de processamento, onde os pontos mais próximos da origem (0, 0) representam um equilíbrio melhor entre estes dois valores, lembrando que somente os pontos não-dominados são considerados. Os parâmetros representados pelo ponto 1 apresentaram o melhor resultado em média de ligações, porém o tempo necessário para alcançar este resultado também foi o maior. Já o ponto 2 obteve uma redução no tempo de processamento. No entanto, a média de ligações caiu em aproximadamente 1 ligação. Entretanto, o ponto 3, apesar de não ter obtido uma média de ligações tão boa quanto o ponto 1, quase igualou o valor do ponto 2. Com relação ao tempo de processamento, o ponto 3 despendeu aproximadamente $1/3$ do tempo comparado com os dois primeiros.

Apesar deste tipo de estudo não indicar um ponto como sendo o melhor entre todos, foram estabelecidos os valores representados pelo ponto 3 como sendo os valores padrão para os parâmetros relativos ao operador de Otimização Parcial.

Após analisar os valores para os operadores, alguns experimentos foram realizados para verificar a utilidade das estratégias implementadas. A primeira delas, chamada Dizimação, apresentou alguns resultados adversos, como pode ser visto na Tabela 10. Como descrito na seção 3.7.1, esperava-se que quando utilizada, esta estratégia auxiliasse o algoritmo a alcançar melhores resultados, já que novas regiões do espaço de busca poderiam ser vasculhadas no decorrer do processo de evolução. Apesar de a diferença ter sido mínima, a média de ligações encontrada foi menor quando utilizada esta estratégia. Outro ponto intrigante refere-se à média de gerações. Era esperado que com a utilização da Dizimação o algoritmo conseguisse obter alguma melhora na população, permitindo que ela evoluísse por mais gerações, o que não aconteceu. Isto pode ter acontecido devido a alguma configuração errônea dos parâmetros relacionados com esta estratégia, como o número de gerações consideradas sem melhora do melhor indivíduo antes de aplicá-la, bem como a porcentagem da população a ser dizimada. Talvez, com uma melhor análise destes parâmetros, esta estratégia possa trazer algum ganho aos resultados obtidos pelo algoritmo. Como uma análise mais aprofundada não foi realizada, decidiu-se não utilizar esta estratégia.

Por fim, os resultados apresentados na Tabela 11 referem-se aos experimentos realizados para a estratégia de Dobramento Progressivo. Percebe-se que, à medida que o número de partes aumenta, os resultados também acompanham este aumento. Isto pode ter acontecido devido ao fato de que com o aumento do número de partes, o tamanho de cada parte é diminuído. Isto faz com que o algoritmo possa se concentrar inicialmente em partes menores e ir avançando na evolução através da agregação de novas partes, formando o dobramento completo ao final das gerações. Porém, isto pode conduzir o dobramento para uma conformação que represente um máximo local, já que as partes que já foram dobradas, que tendem a se compactar, dificilmente serão alteradas. No entanto, o experimento 142 apresentou um resultado pior do que alguns outros experimentos. Com o aumento do número de partes, a partir de certo ponto o tamanho de cada uma delas passa a ser muito pequeno para que a estratégia possa ser eficiente. Isto sugere que existe algum valor intermediário que representa uma maximização da média de ligações e que a partir dele os valores tendam a cair, como no referido experimento.

5.1.2 Seqüências de *benchmark*

Os resultados obtidos pelo *software* GANDALF PRED para estas seqüências, cujo valor máximo de ligações hidrofóbicas não-locais é conhecido, estão apresentados na Tabela 14 juntamente com os resultados obtidos por Unger e Moult (1993c) e König e Dandekar (1999a).

Comparando-se os resultados obtidos para as quatro seqüências mais curtas (20, 24, 25 e 36 resíduos), nota-se que todas as três implementações alcançaram o valor máximo de ligações. Contudo, para a seqüência mais curta, o algoritmo de König e Dandekar encontrou o máximo em todas as rodadas, ao passo que GANDALF PRED encontrou-o em 74% das vezes. No entanto, para a seqüência com 36 resíduos, apesar de todos terem encontrado o valor máximo de ligações e GANDALF PRED tê-lo encontrado menos vezes que o algoritmo de König e Dandekar, a média de ligações obtida por GANDALF PRED é superior à de König e Dandekar.

Para a seqüência de 48 resíduos, observa-se que Unger e Moult não conseguiram obter o máximo de ligações, enquanto tanto König e Dandek quanto GANDALF PRED chegaram a uma conformação com o máximo de ligações. Para a de 50 resíduos, ambas as implementações alcançaram o máximo. Na seqüência de 60 resíduos obteve-se um resultado

interessante. Segundo Unger e Moult (1993c), considerava-se 34 ligações como o número máximo para esta seqüência. Entretanto, GANDALF PRED conseguiu encontrar uma conformação com 35 ligações, superando a implementação anterior.

Considerando-se as duas maiores seqüências (64 e 85 resíduos), observa-se que GANDALF PRED conseguiu obter conformações com o número de ligações muito próximo do máximo considerado, superando claramente as outras implementações.

É importante também ressaltar que à medida que se aumenta o tamanho da seqüência, a diferença na média de ligações obtida por GANDALF PRED em comparação com a implementação de König e Dandekar também vai sendo maior. Isto significa que as conformações encontradas pelo *software* GANDALF PRED são, em média, melhores do que as obtidas pela outra implementação. Isto sugere que o algoritmo possa obter melhores resultados à medida que o tamanho da seqüência aumenta.

Observando-se as conformações da Figura 39 até a Figura 47, percebe-se que a função de *fitness* implementada conseguiu simular corretamente as forças de atração dos resíduos hidrofóbicos, mantendo-os no interior das conformações. Com exceção da Figura 44, todas as outras figuras apresentaram dobramentos onde os resíduos polares tenderam a se manter na parte externa da conformação, de acordo com os princípios utilizados na implementação da função de *fitness*. Nem sempre é possível manter todos os resíduos hidrofílicos externamente e os hidrofóbicos totalmente na parte interna, já que nas proteínas reais estes resíduos hidrofóbicos existentes na parte externa de uma conformação podem contribuir para a formação de novas pontes de hidrogênio com resíduos de outras proteínas e formarem as estruturas quaternárias.

Outra questão interessante trata-se da observação das conformações da Figura 43 e Figura 47. Em ambos os casos, as duas conformações apresentadas são visualmente bem diferentes, apesar de possuírem o mesmo número de ligações. Isto vem confirmar que, pelo menos no modelo 2D HP, pode existir mais de uma conformação nativa para uma seqüência de resíduos de aminoácidos, no caso da Figura 43. Para a Figura 47, este fato confirma a hipótese de que o *fitness landscape* possui vários máximos locais de amplitude similar, tornando ainda mais difícil a tarefa de encontrar o ótimo global, para qualquer tipo de algoritmo.

5.1.3 Seqüências de proteínas reais

No segundo conjunto de testes realizados pelo *software* GANDALF PRED, foram utilizadas 7 proteínas retiradas do PDB. Para estas proteínas, ao contrário do conjunto utilizado nos testes anteriores, não se conhece o valor máximo de ligações hidrofóbicas não-locais para o modelo 2D HP, não sendo, portanto, possível estimar quão próxima uma conformação obtida está de sua conformação nativa.

Uma questão a ser considerada com relação às seqüências de resíduos das proteínas utilizadas nestes testes refere-se ao fato de que, por não terem sido desenvolvidas para serem dobradas em formas compactas, elas tendem a apresentar uma maior dificuldade de se obter o máximo de ligações. Além disto , por existirem vários resíduos polares isolados no meio da seqüência, o algoritmo provavelmente não conseguirá manter somente resíduos hidrofóbicos no interior das conformações e nem somente resíduos polares na parte externa das mesmas.

Observando-se as conformações da Figura 48 até a Figura 54, pode-se perceber que os resíduos hidrofílicos existentes nas proteínas encontram-se bem espalhados na seqüência causando certa dificuldade para o algoritmo em agrupar os resíduos hidrofóbicos no interior das conformações. Porém, com o objetivo de superar esta dificuldade inerente às seqüências testadas, o algoritmo acabou agrupando resíduos polares para que eles não atrapalhassem a formação de outras ligações hidrofóbicas não-locais quando estas fossem possíveis. Pode-se perceber também que, quando possível, o algoritmo conseguiu isolar resíduos polares posicionando-os nos cantos externos das conformações, mantendo os hidrofóbicos livres para formarem outras ligações.

Outra questão a ser comentada refere-se à tendência de as conformações das proteínas mais longas serem formadas por longos segmentos em linha reta. Isto pode ter sido uma estratégia adotada pelo algoritmo, já que, a princípio, uma boa forma de tentar maximizar o número de ligações é tentar ajustar as posições dos resíduos ao longo de segmentos retos.

Apesar das conformações apresentadas não representarem o melhor dobramento para estas proteínas, já que este não é conhecido, e algumas delas possuírem posições internas não ocupadas por nenhum resíduo, pode-se dizer que elas representam dobramentos relativamente bons.

5.2 CONCLUSÕES

O uso do conceito de raio de giração na função de *fitness* é uma das principais contribuições deste trabalho. Com ele, o *fitness landscape* fica mais suave, permitindo que soluções melhores sejam encontradas forçando o posicionamento dos resíduos hidrofóbicos no interior das conformações. Utilizando este conceito na função de *fitness*, duas conformações com o mesmo número de ligações hidrofóbicas não-locais podem ser adequadamente discriminadas.

Outro melhoramento relevante é o operador de Otimização Parcial, que realiza uma busca local em uma parte do indivíduo e, na maioria das vezes, consegue melhorar a conformação como um todo. Apesar de sua utilidade, ele tende a ser computacionalmente custoso à medida que a quantidade de resíduos intermediários aumenta, devendo ser utilizado com parcimônia.

Os operadores de Mutação Sempre Melhor, U-*fold* e Gera *loops* também se mostraram eficientes no processo de evolução. Apesar de sua utilização conjunta não ter apresentado resultados totalmente favoráveis, acredita-se que com algumas alterações na estratégia de utilização dos operadores, estes venham a contribuir mais significativamente para a obtenção de resultados ainda melhores. Isto pode ter acontecido pelo fato de os parâmetros para estes operadores terem sido testados individualmente.

A idéia da estratégia de Dizimação era permitir que o algoritmo pudesse evitar sua estagnação em algum dos vários máximos locais existentes na *fitness landscape*, dando condições para o AG se recuperar de uma possível convergência prematura permitindo, assim, uma exploração mais eficiente do espaço de busca. Porém, os resultados obtidos não foram favoráveis, provavelmente devido à falta de uma análise mais aprofundada da influência dos parâmetros relacionados com esta estratégia.

Um dos objetivos deste trabalho era estabelecer um conjunto de parâmetros que pudesse auxiliar o usuário do algoritmo a obter bons resultados para uma grande parte das instâncias. Após a realização de vários experimentos, um conjunto de parâmetros foi obtido em que houvesse um balanceamento entre desempenho e tempo de processamento.

É importante destacar os resultados obtidos pelo *software* GANDALF PRED. O primeiro conjunto, composto de 9 seqüências cujo valor máximo de ligações é conhecido, indicou que para seqüências curtas, o algoritmo conseguiu atingir os valores ótimos apesar de ser em quantidade menor do que a outra implementação. No entanto, para as seqüências mais longas, as conformações obtidas possuem um número significativamente maior de ligações do

que as outras implementações. Também a média de ligações obtida foi superior às demais. Para todos os casos, os dobramentos encontrados refletem o efeito causado pela função de *fitness*, de modo que os resíduos hidrofóbicos foram posicionados no interior e os polares na parte externa da conformação.

Considerando-se as 7 seqüências de proteínas reais, não é possível afirmar quão próximas as conformações encontradas estão do ótimo global, para o modelo utilizado, mas pode-se observar que as conformações obtidas representam bons dobramentos. Nota-se que devido ao fato de haver resíduos polares isolados na seqüência, a obtenção de uma conformação compacta deve ser difícil. Analisando-se as conformações apresentadas para estas proteínas, percebe-se que o algoritmo foi capaz de agrupar a maioria dos resíduos hidrofóbicos na região central da proteína, posicionando os resíduos polares nas regiões periféricas, quando possível.

A implementação e disponibilização do *software* GANDALF PRED ainda permitirá que novos estudos ou aperfeiçoamentos sejam desenvolvidos nesta área. Assim, os objetivos propostos foram atingidos e os resultados obtidos também foram bastante satisfatórios.

5.3 TRABALHOS FUTUROS

Apesar deste trabalho ter apresentado várias melhorias em comparação com um AG simples, trazendo maior robustez ao algoritmo, novos operadores ainda poderão ser desenvolvidos. Estes operadores podem ser dotados de habilidades “inteligentes” de forma que possam analisar as regiões das conformações e decidir quais tipos de alterações podem ser realizadas para que conformações ainda melhores possam ser obtidas.

Percebeu-se que com a estratégia de somente habilitar o operador de Otimização Parcial após os operadores de U-*fold* e Gera *loops* terem sido automaticamente desativados, não se obtiveram os resultados esperados, de modo que em nenhum dos testes estes três operadores foram aplicados juntamente com o operador Mutação Sempre Melhor. Pode acontecer que, com alguma alteração na estratégia de utilização dos operadores especiais, estes possam ser aplicados todos juntos durante a evolução, de modo a obter resultados ainda melhores. Para isto, o algoritmo deve ser submetido a novos testes e análises.

Para um melhor entendimento da influência de cada parâmetro no processo de evolução e conseqüente definição de parâmetros padrão, seria interessante selecionar conjuntos de valores (da mesma forma como foi feito neste trabalho) e testar todas as

variações possíveis dentro do conjunto de valores definido, inclusive utilizando-se várias proteínas diferentes. Apesar da quantidade de experimentos a serem realizados se tornar muito grande, desta forma poder-se-ia compreender e definir com maior precisão o conjunto de valores padrão.

O modelo 2D HP é muito conhecido. No entanto, o algoritmo poderia ser alterado para encontrar estruturas de proteínas para outros tipos de modelos tridimensionais, tanto simples quanto mais sofisticados, tendendo a se aproximar dos modelos analíticos. Ao sistema GANDALF PRED poderiam ser adicionadas duas características: suporte multiplataforma e implementação de processamento paralelo que visa a diminuição do tempo de processamento e aproveitamento dos recursos de *hardware*, muitas vezes não utilizados e disponíveis em redes de computadores.

ANEXO 1

LISTA DOS AMINOÁCIDOS ESSENCIAIS

Nome	Abreviação		Fórmula Química
Alanina	Ala	A	CH3-CH(NH2)-COOH
Arginina	Arg	R	$HN=C(NH_2)-NH-(CH_2)_3-CH(NH_2)-COOH$
Asparagina	Asn	N	$H_2N-CO-CH_2-CH(NH_2)-COOH$
Aspartato (Ácido Aspártico)	Asp	D	$HOOC-CH_2-CH(NH_2)-COOH$
Cisteína	Cys	C	$HS-CH_2-CH(NH_2)-COOH$
Glutamina (Glutamida)	Gln	Q	$H_2N-CO-(CH_2)_2-CH(NH_2)-COOH$
Glutamato (Ácido Glutâmico)	Glu	E	$HOOC-(CH_2)_2-CH(NH_2)-COOH$
Glicina	Gly	G	NH_2-CH_2-COOH
Histidina	His	H	$NH-CH=N-CH=C-CH_2-CH(NH_2)-COOH$
Isoleucina	Ile	I	$CH_3-CH_2-CH(CH_3)-CH(NH_2)-COOH$
Leucina	Leu	L	$(CH_3)_2-CH-CH_2-CH(NH_2)-COOH$
Lisina	Lys	K	$H_2N-(CH_2)_4-CH(NH_2)-COOH$
Metionina	Met	M	$CH_3-S-(CH_2)_2-CH(NH_2)-COOH$
Fenilalanina	Phe	F	$Ph-CH_2-CH(NH_2)-COOH$
Prolina	Pro	P	$NH-(CH_2)_3-CH-COOH$
Serina	Ser	S	$HO-CH_2-CH(NH_2)-COOH$
Treonina	Thr	T	$CH_3-CH(OH)-CH(NH_2)-COOH$
Triptofano (Triptofana)	Trp	W	$Ph-NH-CH=C-CH_2-CH(NH_2)-COOH$
Tirosina	Tyr	Y	$HO-p-Ph-CH_2-CH(NH_2)-COOH$
Valina	Val	V	$(CH_3)_2-CH-CH(NH_2)-COOH$

REFERÊNCIAS BIBLIOGRÁFICAS

AGARWALA, R.; BATZOGLOU, S.; DACÍK, V.; DECATUR, S. E.; FARACH, M.; HANNENHALLI, S.; SKIENA, S. Local rules for protein folding on a triangular lattice and generalized hydrophobicity in the HP model. **Journal of Computational Biology**, v. 4, p. 275–296, 1997.

ANFINSEN, C. B.; HABER, E.; WHITE, F. H. The kinetics of the formation of native ribonuclease during oxidation of the reduced polypetide domain. **Proceedings of the National Academy of Science of the USA**, v. 47, p. 1309–1314, 1961.

AVBELJ, F.; MOULT, J.; KITSON, D. H.; JAMES, M. N. G.; HAGLER, A. T. Molecular dynamics study of the structure and dynamics of a protein molecule in a crystalline ionic environment, Streptomyces griseus protease A. **Biochemistry**, v. 29, p. 8658–8676, 1990.

BACKOFEN, R. The protein structure prediction problem: a constraint optimization approach using a new lower bound. **Constraints**, v. 6, p. 223–255, 2001.

BARRAL, J. M.; BROADLEY, S. A.; SCHAFFAR, G.; HARTL, F. U. Roles of molecular chaperones in protein misfolding diseases. **Seminars in Cell & Developmental Biology**, v. 15, p. 17–29, 2004.

BEER, F. P.; JOHNSTON, E.R. **Vector Mechanics for Engineers – Statics**. New York, USA: McGraw Hill, 1980.

BERGER, B.; LEIGHTON, F. T. Protein folding in the hydrophobic-hydrophilic (HP) model is NP-complete. **Journal of Computational Biology**, v. 5, p. 27–40, 1998.

BERMAN, H. M.; WESTBROOK, J.; FENG, Z.; GILLILAND, G.; BHAT, T. N.; WEISSIG, H.; SHINDYALOV, I. N.; BOURNE, P. E. The Protein Data Bank. **Nucleic Acids Research**, v. 28, p. 235–242, 2000.

BRANDEN, C.; TOOZE, J. **Introduction to Protein Structure**. 2ª ed. New York: Garland Publishing, 1999.

BROMBERG, S.; DILL, K. A. Side-chain entropy and packing in proteins. **Protein Science**, v. 3, p. 997–1009, 1994.

BROOKS, B. R.; BRUCCOLERI, R. E.; OLAFSON, B. D.; STATES, D. J.; SWAMINATHAN, S.; KARPLUS, M. CHARMM: A Program for Macromolecular Energy, Minimization, and Dynamics Calculations. **Journal of Computational Chemistry**, v.4, p. 187–217, 1983.

CHAN, H. S.; DILL, K. A. “Sequence space soup” of proteins and copolymers. **Journal of Chemical Physics**, v. 95, p. 3775–3787, 1991.

CHAN, H. S.; DILL, K. A. Compact polymers. **Macromolecules**, v. 22, p. 4559–4573, 1989a.

CHAN, H. S.; DILL, K. A. Intrachain loops in polymers: effects of excluded volume. **Journal of Chemical Physics**, v. 90, p. 492–509, 1989b.

CHANDRU, V.; DATTASHARMA, A.; KUMAR, V. S. A. The algorithmics of folding proteins on lattices. **Discrete Applied Mathematics**, v. 127, p. 145–161, 2003.

COOPER, L. R.; CORNE, D. W.; CRABBE, M. J. C. Use of a novel hill-climbing genetic algorithm in protein folding simulations. **Computational Biology and Chemistry**, v. 27, p. 575–580, 2003.

COUNSELL, D. **Bioinformatics of Protein Evolution, Part I**. MRes Biomolecular Sciences Lecture Notes. Disponível em: <http://www.hgmp.mrc.ac.uk/~dcounsel/Mres/MRes2.html>. Acesso em: 16 de abril de 2004a.

COUNSELL, D. **Bioinformatics of Protein Evolution, Part II**. MRes Biomolecular Sciences Lecture Notes. Disponível em: <http://www.hgmp.mrc.ac.uk/~dcounsel/Mres/Mres3.html>. Acesso em: 16 de abril de 2004b.

COVEL, D. G. Lattice model simulations of polypeptide chain folding. **Journal of Molecular Biology**, v. 235, p. 1035–1043, 1994.

CRESCENZI, P.; GOLDMAN, D.; PAPADIMITRIOU, C.; PICCOLBONI, A.; YANNAKAKIS, M. On the complexity of protein folding. **Journal of Computational Biology**, v. 5, p. 423–465, 1998.

CROES, G.A. A method for solving traveling salesman problems. **Operations Research**, v. 6, p. 791–812, 1958.

CSERMELY, P.; SÖTI, C.; KALMAR, E.; PAPP, E.; PATO, B.; VERMES, A.; SREEDHAR, A. S. Molecular chaperones, evolution and medicine. **Journal of Molecular Structure**, v. 666–667, p. 373–380, 2003.

CUI, Y.; CHEN, R. S.; WONG, W. H. Protein folding simulation with genetic algorithm and supersecondary structure constraints. **Proteins: Structure, Function, and Genetics**, v. 31, p. 247–257, 1998.

DANDEKAR, T.; ARGOS, P. Folding the main chain of small proteins with the genetic algorithm. **Journal of Molecular Biology**, v. 236, p. 844–861, 1994.

DANDEKAR, T.; ARGOS, P. *Ab initio* tertiary-fold prediction of helical and non-helical protein chains using a genetic algorithm. **International Journal of Biological Macromolecules**, v. 18, p. 1–4, 1996a.

DANDEKAR, T.; ARGOS, P. Identifying the tertiary fold of small proteins with different topologies from sequence and secondary structure using the genetic algorithm and extended criteria specific for strand regions. **Journal of Molecular Biology**, v. 256, p. 645–660, 1996b.

DAUGHERITY, W. C. A neural-fuzzy system for the protein folding problem. In: Proceedings of the 3rd International Workshop on Industrial Fuzzy Control & Intelligent Systems (IFIS), p. 47–49, 1993.

DAY, R. O.; LAMONT, G. B.; PACHTER, R. Protein structure prediction by applying an evolutionary algorithm. In: **Proceedings of International Parallel and Distributed Processing Symposium (IPDPS)**, p. 155–162, 2003.

DAY, R. O.; ZYDALLIS, J.; LAMONT, G. B. Solving the protein structure prediction problem through a multiobjective genetic algorithm. In: **Technical Proceedings of the 2nd International Conference on Computational Nanoscience and Nanotechnology (ICCN)**, p. 32–35, 2002.

DEB, K. Multi-Objective Optimization using Evolutionary Algorithms. Chichester, UK: John Wiley & Sons, 2001.

DILL, K. A. Theory for the folding and stability of globular proteins. **Biochemistry**, v. 24, p. 1501–1509, 1985.

DILL, K. A.; BROMBERG, S.; YUE, K.; FIEBIG, K. M.; YEE, D. P.; THOMAS, P. D.; CHAN, H. S. Principles of protein folding – a perspective from simple exact models. **Protein Science**, v. 4, p. 561–602, 1995.

DINNER, A. R.; SALI, A.; SMITH, L. J.; DOBSON, C. M.; KARPLUS, M. Understanding protein folding via free-energy surfaces from theory and experiment. **Trends in Biochemical Sciences**, v. 25, p. 331–339, 2000.

DOBSON, C. M. Protein misfolding, evolution and disease. **Trends in Biochemical Sciences**, v. 24, p. 329–332, 1999.

DOBSON, C. M.; EVANS, P. A.; RADFORD, S. E. Understanding how proteins fold: the lysozyme story so far. **Trends in Biochemical Sciences**, v. 19, p. 31–37, 1994.

DOBSON, C. M.; KARPLUS, M. The fundamentals of protein folding: bringing together theory and experiment. **Current Opinion in Structural Biology**, v. 9, p. 92–101, 1999.

DUAN, Y.; KOLLMAN, P. A. Computational protein folding: from lattice to all-atom. **IBM Systems Journal**, v. 40, p. 297–309, 2001.

ELLIS, R. J. Molecular chaperones: avoiding the crowd. **Current Biology**, v. 7, p. R531–R533, 1997.

ELLIS, R. J.; HARTL, F. U. Principles of protein folding in the cellular environment. **Current Opinion in Structural Biology**, v. 9, p. 102–110, 1999.

EYRICH, V. A.; STANDLEY, D. M.; FRIESNER, R. A. Prediction of protein tertiary structure to low resolution: performance for a large and structurally diverse test set. **Journal of Molecular Biology**, v. 288, p. 725–742, 1999.

FELDMAN, D. E.; FRYDMAN, J. Protein folding *in vivo*: the importance of molecular chaperones. **Current Opinion in Structural Biology**, v. 10, p. 26–33, 2000.

FRAENKEL, A. S. Complexity of protein folding. **Bulletin of Mathematical Biology**, v. 55, p. 1199–1210, 1993.

GOLDBERG, D. E. Genetic Algorithms in Search, Optimization, and Machine Learning. Reading, USA: Addison-Wesley, 1989.

GREENWOOD, G. W.; SHIN, J. M.; LEE, B.; FOGEL, G. B. A survey of recent work on evolutionary approaches to the protein folding problem. In: **Proceedings of the 1999 Congress on Evolutionary Computation**, p. 488–495, 1999.

GUEX, N.; DIEMAND, A.; PEITSCH, M. C. Protein modelling for all. **Trends in Biochemical Sciences**, v. 24, p. 364–367, 1999.

HANSMANN, U. H. E.; OKAMOTO, Y. Comparative study of multicanonical and simulated annealing algorithms in the protein folding problem. **Physica A**, v. 212, p. 415–437, 1994.

HAO, M. H.; SCHERAGA, H. A. Computational approach to the statistical mechanics of protein folding. In: **Proceedings of the 1995 ACM/IEEE Conference on Supercomputing**, p. 57–83 , 1995.

HART, W. E.; ISTRAIL, S. Fast protein folding in the Hydrophobic-Hydrofilic model within three-eights of optimal. **Journal of Computational Biology**, v. 3, p. 53–96, 1996.

HART, W. E.; ISTRAIL, S. Lattice and off-lattice side chain models of protein folding: linear time structure prediction better than 86% of optimal. **Journal of Computational Biology**, v. 4, p. 241–259, 1997.

HARTL, F. U. Molecular chaperones in cellular protein folding. **Nature**, v. 381, p. 571–580, 1996.

HENEINE, I. F. **Biofísica Básica**. 1ª ed. São Paulo: Atheneu, 1984.

HEUN, V. Approximate protein folding in the HP side chain model on extended cubic lattices. **Discrete Applied Mathematics**, v. 127, p. 163–177, 2003.

HIROYASU, T.; MIKI, M.; OGURA, S.; AOI, K.; YOSHIDA, T.; OKAMOTO, Y. DONGARRA, J. Energy minimization of protein tertiary structure by parallel simulated annealing using genetic crossover. **IPSJ Transactions on Advanced Computing Systems**, v. 44, 11–27, 2003.

HOLLAND, J.H. **Adaptation in Natural and Artificial Systems**. Cambridge, MA: MIT Press, 1975.

HONIG, B. Protein folding: from the Levinthal paradox to structure prediction. **Journal of Molecular Biology**, v. 293, p. 283–293, 1999.

IRBÄCK, A. Dynamical-parameter algorithms for protein folding. In: Proceedings of the HLRZ Workshop, Monte Carlo Approach to Biopolymers and Protein Folding, p. 98–109, 1998.

JIANG, T.; CUI, Q.; SHI, G.; MA, S. Protein folding simulations of the hydrophobic–hydrophilic model by combining tabu search with genetic algorithms. **Journal of Chemical Physics**, v. 119, p. 4592–4596, 2003.

KAISER JUNIOR, C. E.; LAMONT, G. B.; MERKLE, L. D.; GATES JUNIOR, G. H.; PACHTER, R. Polypeptide structure prediction: real-valued versus binary hybrid genetic algorithms. In: **Proceedings of the 1997 ACM Symposium on Applied Computing**, p. 279–286, 1997.

KARPLUS, M. The Levinthal paradox: yesterday and today. **Folding & Design**, v. 2, p. S69–S75, 1997.

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

KÖNIG, R.; DANDEKAR, T. Improving genetic algorithms for protein folding simulations by systematic crossover. **BioSystems**, v. 50, p. 17–25, 1999a.

KÖNIG, R.; DANDEKAR, T. Refined genetic algorithm simulations to model proteins. **Journal of Molecular Modeling**, v. 5, p. 317–324, 1999b.

KOZA, J.R. Genetic Programming: on the programming of computers by means of natural selection. Cambridge, MA: MIT Press, 1992.

KRASNOGOR, N.; BLACKBURNE, B. P.; HIRST, J. D.; BURKE, E. K. Multimeme algorithms for protein structure prediction. In: **Proceedings of Parallel Problem Solving From Nature, Lecture Notes in Computer Science**, 2002.

KRASNOGOR, N.; HART, W. E.; SMITH, J.; PELTA, D. A. Protein structure prediction with evolutionary algorithms. In: **Proceedings of the International Genetic and Evolutionary Computation Conference (GECCO)**, p. 1596–1601, 1999.

KRASNOGOR, N.; PELTA, D.; LOPEZ, P. E. M.; CANAL, E. Genetic algorithm for the protein folding problem: a critical view. In: **Proceedings of Engineering of Intelligent Systems**, p. 353–360, 1998.

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.

LEE, M. R.; DUAN, Y.; KOLLMAN, P. A. State of the art in studying protein folding and protein structure prediction using molecular dynamics methods. **Journal of Molecular Graphics and Modelling**, v. 19, p. 146–149, 2001.

LEHNINGER, A. L. **Princípios de Bioquímica**. 1ª ed. São Paulo: Sarvier, 1991.

LEHNINGER, A. L.; NELSON, D. L.; COX, M. M. **Principles of Biochemistry**. 2ª ed. New York: Worth Publishers, 2000.

LEONHARD, K.; PRAUSNITZ, J. M.; RADKE, C. J. 3D-lattice Monte Carlo simulations of model proteins. Size effects on folding thermodynamics and kinetics. **Biophysical Chemistry**, v. 106, p. 81–89, 2003.

LI, M. S.; KLIMOV, D. K.; THIRUMALAI, D. Folding in lattice models with side chains. **Computer Physics Communications**, v. 147, p. 625–628, 2002.

LI, H.; HELLING, R.; TANG, C.; WINGREEN, N. Emergence of preferred structures in a simple model of protein folding. **Science**, v. 273, p. 666–669, 1996.

LI, H.; TANG, C.; WINGREEN, N. S. Nature of driving force for protein folding: a result from analyzing the statistical potential. **Physical Review Letters**, v. 79, p. 765–768, 1997.

LIANG, F.; WONG, W. H. Evolutionary Monte Carlo for protein folding simulations. **Journal of Chemical Physics**, v. 115, p. 3374–3380, 2001.

LYNGSØ, R. B.; PEDERSEN, C. N. S. Protein folding in the 2D HP model. In: Proceedings of the 1st Journées Ouvertes: Biologie, Informatique et Mathématiques (JOBIM), 2000.

MAURI, G.; PAVESI, G.; PICCOLBONI, A. Approximation algorithms for protein folding prediction. In: **Proceedings of the 10th Annual Symposium on Discrete Algorithms (SODA)**, p. S945–S946, 1999.

MERKLE, L. D.; GAULKE, R. L.; LAMONT, G. B.; GATES, G. H.; PACHTER, R. Hybrid genetic algorithms for polypeptide energy minimization. In: **Proceedings of the 1996 Symposium on Applied Computing**, 1996.

MOGK, A.; BUKAU, B. Molecular chaperones: structure of a protein disaggregase. **Current Biology**, v. 14, p. R78–R80, 2004.

NAKAMURA, H. K.; SASAKI, T. N.; SASAI, M. Strange kinetics and complex energy landscapes in a lattice model of protein folding. **Chemical Physics Letters**, v. 347, p. 247–254, 2001.

NAYAK, A.; SINCLAIR, A.; ZWICK, U. Spatial codes and the hardness of string folding problems. In: **Proceedings of the 9th Annual Symposium on Discrete Algorithms (SODA)**, p. 639–648, 1998.

NEWMAN, A. A new algorithm for protein folding in the HP model. In: **Proceedings of the 13th Annual Symposium on Discrete Algorithms (SODA)**, p. 876–884, 2002.

NGO, J. T.; MARKS, J. Computational complexity of a problem in molecular structure prediction. **Protein Engineering**, v. 5, p. 313–321, 1992.

NGO, J. T.; MARKS, J.; KARPLUS, M. Computational complexity, protein structure prediction, and the Levinthal paradox. In: MERZ JUNIOR, K.; LeGrand, S. (eds.) **The Protein Folding Problem and Terciary Structure Prediction**. Boston: Birkhäuser, p. 433–506, 1994.

NUNES, N. L.; CHEN, K.; HUTCHINSON, J. S. Flexible lattice model to study protein folding. **Journal of Physical Chemistry**, v. 100, p. 10443–10449, 1996.

O'TOOLE, E. M.; PANAGIOTOPOULOS, A. Z. Monte Carlo simulation of folding transitions of simple model proteins using a chain growth algorithm. **Journal of Chemical Physics**, v. 97, p. 8644–8652, 1992.

OKAMOTO, Y. Towards the prediction of protein tertiary structures from first principles. **Physica A**, v. 254, p. 7–14, 1998.

ORTIZ, A. R.; KOLINSKI, A.; SKOLNICK, J. Fold assembly of small proteins using Monte Carlo simulations driven by restraints derived from multiple sequence alignments. **Journal of Molecular Biology**, v. 277, p. 419–448, 1998.

OSGUTHORPE, D. J. *Ab initio* protein folding. **Current Opinion in Structural Biology**, v. 10, p. 146–152, 2000.

OSTROVSKY, B.; CROOKS, G.; SMITH, M. A.; BAR-YAM, Y. Cellular automata for polymer simulation with application to polymer melts and polymer collapse including implications for protein folding. **Parallel Computing**, v. 27, p. 613–641, 2001.

PAPANDREOU, N.; KANEHISA, M.; CHOMILIER, J. Folding the human protein FKBP: lattice Monte Carlo simulations. **Comptes Rendus de l'Académie des Sciences Paris, Sciences de la Vie**, v. 321, p. 835–843, 1998.

PATTON, A. L.; PUNCH III, W. F.; GOODMAN, E. D. A standard GA approach to native protein conformation prediction. In: **Proceedings of the 6th International Conference on Genetic Algorithms**, p. 574–581, 1995.

PEDERSEN, C. N. S. **Algorithms in Computational Biology**. PhD Thesis, Department of Computer Science. University of Aarhus, Denmark, 2000.

PEDERSEN, J. T.; MOULT, J. Protein folding simulations with genetic algorithms and a detailed molecular description. **Journal of Molecular Biology**, v. 269, p. 240–259, 1997.

PICCOLBONI, A.; MAURI, G. Application of evolutionary algorithms to protein folding prediction. In: **Selected Papers from the 3rd European Conference on Artificial Evolution, Lecture Notes in Computer Science**, p. 123–136, 1998.

PONTIN, C. P.; RUSSELL, R. R. The natural history of protein domains. **Annual Review of Biophysics and Biomolecular Structure**, v. 31, p. 45–71, 2002.

RABOW, A. A.; SCHERAGA, H. A. Lattice neural network minimization: application of neural network optimization for locating the global-minimum conformations of proteins. **Journal of Molecular Biology**, v. 232, p. 1157–1168, 1993.

RABOW, A. A.; SCHERAGA, H. A. Improved genetic algorithm for the protein folding problem by use of a cartesian combination operator. **Protein Science**, v. 5, p. 1800–1815, 1996.

RADER, A. J.; BAHAR, I. Folding core predictions from network models of proteins. **Polymer**, v. 45, p. 659–668, 2004.

ROOMAN, M.; KOCHER, J. P.; WODAK, S. Prediction of protein backbone conformation based on seven structural assignments. **Journal of Molecular Biology**, v. 221, p. 961–979, 1991.

SALI, A.; SHAKHNOVICH, E.; KARPLUS, M. Kinetics of protein folding: a lattice model study of the requirements for folding to the native state. **Journal of Molecular Biology**, v. 235, p. 1614–1636, 1994.

SCAPIN, M. P.; LOPES, H. S. Protein structure prediction using an enhanced genetic algorithm for the 2D HP model. In: **Proceedings of 3rd Brazilian Workshop on Bioinformatics, Brasília (DF), CD-ROM, 2004.**

SCHULZE-KREMER, S.; TIEDEMANN, U. Parameterizing genetic algorithms for protein folding simulation. In: **Proceedings of the 27th Annual Hawaii International Conference on System Sciences**, p. 345–354, 1994.

SHAKHNOVICH, E. I.; GUTIN, A. M. Engineering of stable and fast-folding sequences of model proteins. **Proceedings of the National Academy of Science of the USA**, v. 90, p. 7195–7199, 1993.

SHMYGELSKA, A.; HERNÁNDEZ, R. A.; HOOS, H. H. An ant colony optimization algorithm for the 2D HP protein folding problem. In: **Proceedings of the 3rd International Workshop (ANTS), Lecture Notes in Computer Science**, v. 2463, p. 40–55, 2002.

SHMYGELSKA, A.; HOOS, H. H. An improved ant colony optimisation algorithm for the 2D HP protein folding problem. In: **Proceedings of the 16th Canadian Conference on Artificial Intelligence**, 2003.

SIMON, I.; FISER, A.; TUSNÁDY, G. E. Predicting protein conformation by statistical methods. **Biochimica et Biophysica Acta**, v. 1549, p. 123–136, 2001.

SOCCI, N. D.; ONUCHIC, J. N. Folding kinetics of protein-like heteropolymers. **Journal of Chemical Physics**, v. 101, p. 1519–1528, 1994.

SRINIVASAN, R.; ROSE, G. D. *Ab initio* prediction of protein structure using LINUS. **Proteins: Structure, Function, and Genetics**, v. 47, p. 489–495, 2002.

STANSFIELD, W. D. **Genética**. 2ª ed. São Paulo: McGraw-Hill do Brasil, 1985.

STEIPE, B. Protein Design Concepts. In: SCHLEYER, P. V. R.; ALLINGER, N. L.; CLARK, T.; GASTEIGER, J.; KOLLMAN, P. A.; SCHAEFER III, H. F.; SCHREINER, P. R. (eds.) **The Encyclopedia of Computational Chemistry**. Chichester: John Wiley & Sons, p. 2168–2185, 1998.

SUNDARARAJAN, V.; EILS, R. Evolving protein structures through genetic algorithms. In: **Poster in HPC Asia**, 2002.

TAKAHASHI, O.; KITA, H.; KOBAYASHI, S. Protein folding by a hierarchical genetic algorithm. In: **Proceedings of the 4th International Symposium on Artificial Life and Robotics (AROB)**, 1999.

TANG, C. Simple models of the protein folding problem. **Physica A**, v. 288, p. 31–48, 2000.

THOMAS, P. D.; DILL, K. A. Local and nonlocal interactions in globular proteins and mechanisms of alcohol denaturation. **Protein Science**, v. 2, p. 2050–2065, 1993.

THOMASSON, W. A. B. Unraveling the mystery of protein folding. **Breakthroughs in Bioscience**. FASEB Office of Public Affairs, 2001.

UFSC – UNIVERSIDADE FEDERAL DE SANTA CATARINA. Revista Eletrônica do Departamento de Química. **O Mundo das Proteínas.** Disponível em: <http://qmc.ufsc.br/qmcweb/artigos/proteinas.html>. Acesso em: 23 de fevereiro de 2004.

UNGER, R.; MOULT, J. A genetic algorithm for three dimensional protein folding simulations. In: **Proceedings of the 5th Annual International Conference on Genetic Algorithms**, p. 581–588, 1993a.

UNGER, R.; MOULT, J. Finding the lowest free energy conformation of a protein is a NP-hard problem: proof and implications. **Bulletin of Mathematical Biology**, v. 55, p. 1183–1198, 1993b.

UNGER, R.; MOULT, J. Genetic algorithms for protein folding simulations. **Journal of Molecular Biology**, v. 231, p. 75–81, 1993c.

UNGER, R.; MOULT, J. On the applicability of genetic algorithms to protein folding. In: **26th Hawaii International Conference on System Sciences**, v. I, p. 715–725, 1993d.

XU, Y.; XU, D.; UBERBACHER, E. C. A new method for modeling and solving the protein fold recognition problem. In: **Proceedings of the 2nd Annual International Conference on Research in Computational Molecular Biology (RECOMB)**, p. 285–292, 1998.

YESYLEVSKYY, S. O.; DEMCHENKO, A. P. Towards realistic description of collective motions in the lattice protein folding models. **Biophysical Chemistry**, v. 109, p. 17–40, 2004.

YUE, K.; DILL, K. A. Sequence-structure relationships in proteins and copolymers. **Physical Review E**, v. 48, p. 2267–2278, 1993.

RESUMO:

Este trabalho propõe a utilização da técnica de computação evolucionária conhecida como algoritmos genéticos (AGs) na predição da estrutura de proteínas para o modelo 2D HP. A metodologia tem como principal proposta a utilização uma função de *fitness* melhorada, que utiliza o conceito de raio de giração. Operadores genéticos especiais foram desenvolvidos e acrescentados aos comumente usados em AG, além de novas estratégias utilizadas para auxiliar o algoritmo no processo de busca de conformações de proteínas. Estas modificações levaram ao desenvolvimento de um sistema de *software* com diversos recursos gráficos e apresentação de relatórios dos resultados, denominado GANDALF PRED. Uma certa quantidade de experimentos foi realizada com o objetivo de avaliar a influência parâmetros do AG no resultado obtido. Foram realizados dois conjuntos de testes para avaliar a metodologia proposta. O primeiro utilizou 9 seqüências de resíduos, manualmente definidas, cujos máximos de ligações são conhecidos e comprimento variando de 20 a 85 resíduos. Os resultados obtidos foram comparados com duas outras implementações encontradas na literatura. No segundo, 7 proteínas com características globulares foram escolhidas do PDB e traduzidas para o modelo HP. Seus comprimentos variam de 288 a 842 resíduos. Seus resultados foram apresentados e discutidos, já que nenhuma comparação pôde ser realizada. Para ambos os casos de teste, as conformações encontradas podem ser consideradas bons dobramentos.

PALAVRAS-CHAVE
Algoritmos genéticos, dobramento de proteínas, predição de estruturas, modelo 2D HP.

ÁREA/SUB-ÁREA DE CONHECIMENTO
1.03.03.04–9 Sistemas de Informação

2.08.04.00–8 Biologia Molecular

2005

N°: 359