DEPARTAMENTO DE ENGENHARIA INFORMÁTICA
FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

PREVISÃO DA CONFORMAÇÃO DE PROTEÍNAS: UMA ABORDAGEM EVOLUCIONÁRIA

Ricardo Manuel da Conceição Rodrigues

EVOLUTIONARY AND COMPLEX SYSTEMS GROUP
CENTRO DE INFORMÁTICA E SISTEMAS DA UNIVERSIDADE DE COIMBRA
COIMBRA, PORTUGAL
DEZEMBRO, 2007

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA
FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

PREVISÃO DA CONFORMAÇÃO DE PROTEÍNAS: UMA ABORDAGEM EVOLUCIONÁRIA

Ricardo Manuel da Conceição Rodrigues

Dissertação submetida à Universidade de Coimbra para a obtenção do grau de Mestre em Informática e Sistemas, sob orientação do **Prof. Doutor Ernesto Costa**, Professor Catedrático de Nomeação Definitiva do Departamento de Engenharia Informática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra

À minha Família e Amigos.

Índice

Lista de Tabelas

Lista de Figuras

Lista de Algoritmos

Resumo

O tema desta dissertação é a previsão da conformação tridimensional de proteínas, através da utilização de um modelo simples, bidimensional, como forma de validação de uma abordagem evolucionária com algoritmos genéticos.

A previsão da conformação de proteínas é um problema bem definido dentro da área da biologia computacional, tendo sido estudado nas últimas duas décadas, devido à relação existente entre a forma de uma proteína e a função que esta desempenha. Há especial interesse nesta área, em particular, por parte da indústria farmacêutica (tanto para um melhor conhecimento de determinadas doenças, como para a produção de fármacos que combatam essas mesmas doenças) e, de um modo mais geral, por parte das ciências da vida.

As soluções informáticas para este problema recorrem a algoritmos de procura. O problema é, contudo, bastante complexo, devido à dimensão e características do espaço de procura envolvido, obrigando à utilização de soluções heurísticas e à necessidade de uma aproximação à solução óptima. Tal acontece porque, muitas vezes, é virtualmente impossível determinar a solução óptima.

Das várias abordagens possíveis para este problema, a escolhida foi a computação evolucionária, por recurso a algoritmos genéticos. Os algoritmos genéticos são das técnicas actuais com maior sucesso para problemas em que existe um espaço imenso de soluções candidatas e é necessário fazer uma escolha em função da qualidade dessas soluções, através de um processo de avaliação individual.

Sendo, no entanto, o problema real da previsão da conformação de proteínas bastante complexo, optou-se pela utilização de um modelo inicial simples e já bem documentado que, mesmo assim, não retira a característica NP-difícil do problema: o Modelo HP. Este modelo apenas toma em conta uma das propriedades dos aminoácidos que compõem as proteínas: serem hidrofóbicos ou polares, diminuindo assim parcialmente a complexidade do problema em questão.

A aplicação de algoritmos genéticos a este tipo de problemas mostrou já resultados promissores, havendo no entanto ainda espaço para a utilização de técnicas específicas para o problema em questão. Assim, neste trabalho, propõe-se o recurso à reparação de indivíduos, não se descartando soluções que, embora inválidas, possam encontrar-se perto de uma solução com boas características. No algoritmo genético usado pela abordagem proposta foram ainda incorporados diversos mecanismos, entre os quais merecem especial destaque a utilização de vários operadores de macromutação e a utilização de taxas dinâmicas de variação. Todos estes mecanismos provaram ser capazes de fornecer bons resultados, embora justifiquem ainda melhoramentos a efectuar em trabalhos futuros, incluindo a sua aplicação mesmo fora do domínio dos algoritmos genéticos.

Abstract

The subject of this dissertation is the protein three-dimensional structure prediction by means of a simple two-dimensional model, as a way of validating an evolutionary approach with genetic algorithms.

Protein structure prediction is a well-defined problem within the domain of computational biology, or bioinformatics, being an object of study in the last two decades, due to the existing relation between the shape of a protein and the functions performed by it. Pharmaceutical industries, in particular, and life sciences, in general, have special interest in this area, as a way to enhance the knowledge about some diseases, as to make pharmaceutical products to fight those same diseases.

The computational solutions to this problem make use of search algorithms. However, the problem is too complex due to the dimension of the search space, obliging to the use of heuristic techniques and the necessity of an approximation to an optimal solution. Many times, that is due to being virtually impossible to find an optimal solution.

Of the possible approaches to this problem, the chosen one was evolutionary computation with the specific use of genetic algorithms. Genetic algorithms have been one of the current techniques with greater success in problems with a vast space of candidate solutions and where there is a need to find out which one is the best, in function of their quality, through an individual evaluation process.

But, as protein structure prediction is too complex, a well-documented simple model that still retains the NP-hard characteristic of the problem has been used: the HP model. This specific model only works with the hydrophobic-polar properties of the amino acids that compose proteins, decreasing the complexity of the problem.

The use of genetic algorithms in this kind of problems has already shown a good outcome, but there is still an improvement margin with the use of problem specific techniques such as the individual's repair mechanism here shown, which doesn't discard invalid solutions but, instead, tries to correct them. In the genetic algorithm

used in the proposed approach there have been incorporated also mechanisms such as the use of dynamic variation probabilities and macromutation operators. These mechanisms have proven being capable of providing good results in the studied problem, although still justifying improvements to be made in future works, or even its application outside the domain of genetic algorithms.

Agradecimentos

Gostaria de agradecer ao Prof. Doutor Ernesto Costa, meu orientador, pelas suas muitas e constantes sugestões durante todo o trabalho realizado, tendo-me permitido um primeiro contacto com o tema desta dissertação e, também, apoiado sempre durante o aprofundamento do mesmo e realização da dissertação.

Devo também apresentar os meus agradecimentos aos elementos do *Evolutionary and Complex Systems Group*, ECOS, parte do Centro de Informática e Sistemas da Universidade de Coimbra, CISUC, no qual me integro, que foram fonte de crítica e apoio para um melhor desempenho da minha parte, apresentando, por vezes, o caminho a seguir, sempre úteis nos seus contributos.

Um especial agradecimento é dirigido aos meus pais por toda a sua paciência e apoio. Sem eles, todo este trabalho, e tudo quanto o precedeu, não teria sido possível.

Quero também agradecer aos amigos que me têm acompanhado na Lusa Atenas, em especial ao Marco Veloso, ao Nuno Gil, ao Gonçalo Faria, ao João Cunha e ao Miguel Silva, por todo o tempo que passamos juntos e companheirismo demonstrado ao longo da última dezena de anos.

Finalmente, quero agradecer à minha mui querida Joana, por todos os motivos e mais alguns, mas em especial pelo seu apoio e força na recta final da escrita desta dissertação.

Coimbra, Portugal — Ricardo Rodrigues
Dezembro, 2007

Capítulo 1

Introdução

Este documento surge no âmbito do Mestrado em Informática e Sistemas da Universidade de Coimbra, centrado especificamente na área da Computação Evolucionária, e descreve o trabalho realizado durante o mesmo e as conclusões obtidas.

1.1 Motivação

Uma das razões de ser da investigação subjacente à biologia computacional, especialmente no tema abordado nesta dissertação, pode ser definida pela seguinte frase de Francis Crick,[1] que traduz o chamado Dogma Central da Biologia:

"*DNA makes RNA, RNA makes protein, and proteins make us.*"

Esta dissertação centra-se concretamente nas proteínas e nas sequências de aminoácidos que as constituem — o ADN é transcrito para ARN, e este é traduzido nos vários aminoácidos que constituem uma proteína (ver Fig. 1.1). Em específico, uma proteína é definida pela sequência de aminoácidos que a constitui, sendo a sequência de aminoácidos, assim, um importante objecto de estudo.

ADN → ARN → Proteínas

Figura 1.1: Dogma central da biologia

Sabe-se que as proteínas são importantes. Por exemplo, é-nos dito quotidianamente em campanhas publicitárias de vários produtos alimentares que estes são ricos

[1] Francis Crick, juntamente com James Watson, foi o cientista responsável pela descoberta da forma de dupla hélice da molécula de ADN, em 1953.

em proteínas que fortalecem o organismo humano no dia-a-dia. Há esta noção de que as proteínas são importantes pelas funções que desempenham. De facto, as proteínas desempenham um papel crucial em virtualmente todos os processos biológicos, determinando o padrão das transformações químicas nas células. Note-se, por exemplo, que quase todos os catalisadores em sistemas biológicos são proteínas, chamadas enzimas. As proteínas são responsáveis por uma vasta gama de funções, nomeadamente [Str95]:

- **catálise enzimática** — assistência à realização das reacções químicas em sistemas biológicos;
- **transporte e armazenamento de moléculas** — e.g., o transporte de oxigénio no sangue pela mioglobina;
- **movimentação coordenada** — e.g., as contracções musculares do coração, que necessitam da sincronização das células do músculo cardíaco;
- **sustentação mecânica** — e.g., a queratina das unhas, pele e cabelo, que tem como função fornecer resistência aos mesmos;
- **protecção imunitária** — e.g., a imunoglobulina, gerada pelos linfócitos, que tem como função reconhecer e neutralizar corpos estranhos ao organismo;
- **geração e transmissão de impulsos nervosos** — a resposta de células nervosas a impulsos é feita por intermédio de proteínas receptoras específicas; e.g., a rodopsina é a proteína receptora nos bastonetes da retina;
- **controlo do metabolismo, do crescimento e da diferenciação celular** — algumas proteínas ajudam a regular a actividade celular ou fisiológica; e.g., a insulina, que regula o metabolismo dos açúcares, ou a hormona de crescimento da hipófise.

Sabe-se que a função das proteínas está profundamente dependente da sua estrutura tridimensional — i.e., a forma espacial que a macromolécula adopta num determinado ambiente —, tendo todas as proteínas uma forma tridimensional nativa. As proteínas adoptam intrinsecamente determinada forma em função dos aminoácidos que as constituem, da ordem pela qual estes se encontram e das relações entre estes.

Assim, conhecendo-se *a priori* a estrutura tridimensional de uma proteína, será permitido determinar qual a sua função específica. Isto reveste-se de especial importância no (re)conhecimento de doenças, na medida em que conhecendo as proteínas

envolvidas nas doenças será possível apontar possíveis medidas para combater as mesmas — é um caso de especial interesse o estudo das doenças do foro genético. O problema também pode assumir outra forma: sabendo-se que é necessária a síntese de uma proteína com determinadas características (e funções), como descobrir qual a proteína que assumirá a estrutura tridimensional pretendida, e depois criar um novo fármaco baseado nessa informação? Esta questão pode ser bem justificada se se tiver em conta que, segundo relatórios de indústrias farmacêuticas, por cada novo fármaco lançado no mercado existem cerca de 10.000 candidatos para testes pré-clínicos, podendo custar todo este processo de €750 a €800 milhões, onde 75% deste valor pode ser atribuído a falhas ao longo do processo de desenvolvimento.

As proteínas são de tal forma importantes na regulação das funções das células, que muitos acreditam que o proteoma — o conjunto de proteínas que pode ser encontrado numa dada célula — passará a ser, se não for já, um dos principais objectos de estudo da bioinformática [Sea00].

1.2 Problema

Para responder à questão de qual é a estrutura tridimensional de uma proteína, existem duas abordagens bem distintas: através de métodos laboratoriais (*in vitro*) e/ou através de métodos informáticos (*in silico*). Os métodos laboratoriais mais conhecidos são a Ressonância Magnética Nuclear e a Cristalografia de Raios X. Estes métodos são, no entanto, bastante dispendiosos, seja em termos monetários, seja em termos de tempo; além de que, para algumas proteínas, é virtualmente impossível obter desta forma a sua estrutura tridimensional, devido à sua complexidade. Estas limitações levaram à criação de técnicas informáticas.

Com técnicas informáticas entra-se no domínio da previsão — i.e., procura-se encontrar um bom modelo, mas não existem garantias que este seja o mais correcto —, mas, por outro lado, podem-se obter modelos de uma forma mais rápida e menos dispendiosa, nomeadamente no domínio da simulação. A conformação pode ser definida livremente como o processo de uma proteína adquirir a sua forma. A previsão da conformação de proteínas é definida como a tentativa de descobrir a forma que uma dada proteína vai adoptar em condições ambientais ideais.

Existem três abordagens possíveis à Previsão da Conformação de Proteínas: Modelação Comparativa, Reconhecimento de Dobras,[2] e Previsão *ab initio* [SL02], as

[2]Reconhecimento de dobras, a partir da expressão inglesa *fold recognition*. Outra tradução

quais se descrevem sumariamente de seguida:

- na **Modelação Comparativa** é utilizado o facto de que proteínas relacionadas evolutivamente terão também, dependendo da percentagem de aminoácidos em comum e da ordem em que estes se encontram, estruturas similares;

- no **Reconhecimento de Dobras** — i.e., reconhecimento de padrões estruturais que as proteínas assumem — usa-se uma função de avaliação que compara padrões já conhecidos (e as suas sequências de aminoácidos) com as sequências das proteínas a explorar. Neste método dá-se maior relevância à comparação de excertos do que ao todo, permitindo a construção do modelo final com vários padrões, correspondente aos vários excertos da proteína;

- na **Previsão *ab initio***, ao contrário das outras duas abordagens, não são utilizadas outras proteínas para obter a conformação da proteína em estudo. É apenas utilizada a sequência de aminoácidos da proteína para esse efeito, bem como propriedades já conhecidas que contribuem para o conformar de proteínas — e.g., propriedades hidrofóbicas e polares dos aminoácidos, ligações químicas covalentes, e forças de *van der Waals*.

Os três métodos só podem ser validados comparando-se os resultados obtidos com resultados experimentais, mas é o terceiro método aquele que oferece maiores possibilidades na obtenção de prováveis soluções, uma vez que funciona de forma autónoma — i.e., não é necessário o acesso a outras proteínas para obter resultados. A unidade de medida para verificação da aproximação dos modelos previstos ao modelo real é o Ångström (Å), que é equivalente a 0,1 nanómetros e é utilizado para medir a distância média entre esqueletos ou *carbonos alfa* (C_α) de proteínas sobrepostas (utilizando-se a raiz quadrada do desvio médio quadrático entre vários pontos dos modelos em comparação). Quanto menor for o desvio, melhor a solução encontrada. Regra geral, numa boa solução, o desvio deve-se encontrar na casa das unidades.

Em qualquer dos casos, nenhum deles é totalmente fiável, e a Previsão *ab initio* ainda é objecto de muito estudo, uma vez que ainda não se conhecem todos os detalhes dos factores que influenciam a conformação das proteínas.

Dentro das previsões *ab initio*, devido à grande complexidade de conjugação de todos os factores envolvidos [Dil90], surgiram modelos simples, tais como o proposto

possível, mas com menos expressão, é "reconhecimento de enrolamentos".

por K. Dill e K. Lau no seu artigo seminal sobre o Modelo HP [LD89], que foca apenas as propriedades hidrofóbicas e polares dos aminoácidos. Neste modelo ficam de fora as ligações electro-estáticas presentes nas proteínas, ligações de hidrogénio e interacções *van der Waals*, interacções locais (dependentes das propriedades intrínsecas aos aminoácidos em questão) e o efeito hidrofóbico sobre alguns dos aminoácidos, para não mencionar a importância do substracto onde se encontra a proteína estudo. Apesar desta simplificação, é sabido que o efeito hidrofóbico é o parâmetro mais relevante na conformação de proteínas.

O Modelo HP, seja ele bidimensional ou tridimensional, é um modelo computacional simplificado para investigação da conformação de proteínas, devido à complexidade dos aspectos a ter em conta (e já mencionados). Neste modelo, os aminoácidos são representados como contas num reticulado plano (ou cúbico, se for utilizado o modelo tridimensional), com distâncias predefinidas entre elas, e caracterizadas como hidrofóbicas *(H)* ou polares *(P)*. Os aminoácidos hidrofóbicos procuram afastar-se do contacto com soluções aquosas, procurando o centro da proteína, ao passo que os aminoácidos polares procuram estar em contacto com a solução aquosa onde se encontra a proteína. Assim, neste modelo é procurada uma conformação com um mínimo de energia, onde os aminoácidos hidrofóbicos se encontrem juntos no interior da proteína, e os aminoácidos polares no exterior da proteína.

As várias soluções para os modelos propostos para determinada proteína são avaliadas por uma função de energia, que atribui pontos negativos por cada interacção entre (ou por cada par de) aminoácidos hidrofóbicos não adjacentes. Quanto menor o número devolvido pela função de avaliação sobre um modelo, melhor a qualidade do mesmo.

1.3 Objectivo do Trabalho

Nesta dissertação é abordado o método da previsão *ab initio*, utilizando o Modelo HP 2D. O trabalho é elaborado dentro da área da Computação Evolucionária, mais especificamente com recurso a Algoritmos Genéticos. A escolha por este método prendeu-se com diversos factores, entre os quais se destacam:

- ser uma técnica relativamente recente, nomeadamente na sua aplicação à previsão da conformação de proteínas;
- apresentar soluções bastante próximas do óptimo no domínio da previsão;

- o modelo HP, apesar de simples, ser uma boa base de testes para a utilização de algoritmos genéticos no domínio da previsão da estrutura de proteínas, podendo estes depois serem aplicados em modelos mais próximos da realidade.

De um modo mais concreto, pretende-se demonstrar a adequação de uma nova abordagem evolucionária à conformação de proteínas. Para concretizar este objectivo, foi desenvolvido um novo algoritmo genético, cujo desempenho foi testado experimentalmente com recurso a várias sequências de teste padrão. Assim, os vários objectivos deste trabalho passaram por:

- validar a possibilidade de uma nova abordagem evolucionária onde possam ser acrescentados/usados mecanismos que melhorem a abordagem clássica com algoritmos genéticos, mas sem a desvirtuar grandemente;

- transpor, para o problema em estudo, técnicas como a utilização de taxas dinâmicas de variação, que ainda não tinham sido (tanto quanto se sabe) usadas neste problema;

- utilizar operadores de macromutação (alguns já existentes e outros novos), procurando observar o seu efeito sofre os indivíduos e sobre a população em geral;

- propor um mecanismo que procure aproveitar parte dos indivíduos inválidos gerados (que constituem a esmagadora maioria da descendência) através da sua reparação, permitindo a recuperação de parte do seu genótipo ao mesmo tempo que se reduz o tempo de espera para a constituição de uma população (ou descendência) de indivíduos (todos) válidos.

1.4 Contributos Originais

Para além da experimentação com operadores genéticos ao nível da variação — alguns já conhecidos e outros novos — e da aplicação de taxas variáveis de utilização a aplicar a cada um deles, no âmbito da definição do algoritmo genético que se encontra no coração do programa de suporte ao estudo subjacente a esta dissertação, um aspecto digno de nota especial é a utilização de um mecanismo para reparação de indivíduos (soluções candidatas para o problema em questão), que representam conformações possíveis para uma dada proteína.

Devido à natureza do problema em estudo, e também devido à elevada geração de indivíduos inválidos, um dos aspectos a evitar é a convergência prematura dos indivíduos da população para um mínimo local. A estratégia adoptada para evitar esta convergência, apesar de já ser conhecida, não é habitual neste tipo de problemas de previsão da conformação de proteínas num modelo bidimensional: os valores adoptados para as probabilidades de mutação e recombinação são ajustáveis, de forma predefinida, durante a execução do programa. Adapta-se assim, de forma incremental ou decremental, a probabilidade de recombinação e as probabilidades de mutação e macromutação, com o intuito de gerar maior diversidade quando a população tende a estagnar. As taxas de variação voltam a assumir o seu valor inicial mais tarde, quando houver evolução dos melhores indivíduos da população. Quando tal acontece, regra geral, também a diversidade dessa população aumenta.

Através da observação e análise de resultados, foi possível notar que, por vezes, indivíduos promissores eram descartados devido a pequenas falhas — e.g., dois aminoácidos a ocupar a mesma posição numa ponta exterior. Observou-se também que, muitas das vezes, corrigir estas pequenas falhas seria relativamente simples e proporcionaria avanços na procura do melhor indivíduo que solucionasse o problema em estudo, ou se aproximasse da solução óptima. Com isso em mente, foi desenvolvido um mecanismo de reparação que, dado um indivíduo *inválido*, procura iterativamente as falhas existentes na estrutura desse indivíduo e tenta depois repará-las. É utilizado o termo "tenta" porque se optou por limitar o número de iterações para reparar determinado indivíduo, uma vez que, se existirem demasiadas reparações, a probabilidade do indivíduo reparado ser melhor que o indivíduo não reparado não aumenta de forma notória. E também tem de ser tomado em conta o tempo que demora a reparar o indivíduo, que aumenta com o número de iterações necessárias. Na forma actual do mecanismo de reparação, não é de todo improvável a ocorrência de ciclos durante a reparação — a reparação num determinado local poder dar origem a uma falha noutro local e vice-versa — sem que progressos sejam verificáveis, sendo este facto minorado se existir um número de tentativas limitado.

O mecanismo de reparação adoptado começa, de forma aleatória, por umas das pontas da sequência em estudo; procura um ponto de intersecção entre dois aminoácidos dessa cadeia; e antes da intersecção (dependendo a posição onde é aplicada a reparação do tipo de falha apresentada), altera a direcção codificada no genótipo do indivíduo, criando um *novo* e testando-o. Assim acontece sucessivamente até o novo indivíduo gerado ser válido ou o número máximo de iterações permitidos para uma

reparação tiver sido atingido.

1.5 Estrutura da Dissertação

Esta dissertação encontra-se dividida por vários capítulos, e inicia-se com esta introdução (**Capítulo 1**).

De seguida, no **Capítulo 2**, fornece-se a uma introdução aos conceitos biológicos e informáticos indispensáveis para uma abordagem ao problema aqui estudado.

O **Capítulo 3** apresenta em detalhe o modelo HP e o estado da arte, focando os trabalhos que revelaram melhores resultados, desde os que surgiram logo após a proposta do modelo até aos mais recentes.

É depois apresentada, no **Capítulo 4**, a estrutura do algoritmo genético utilizado no programa que serviu de base aos testes e forneceu os resultados discutidos nesta dissertação. São descritos aspectos chave como a estrutura base do algoritmo, a caracterização dos indivíduos e da população, a função de avaliação, a selecção de progenitores e os operadores genéticos utilizados, para além da parametrização do programa.

No **Capítulo 5** apresentam-se os resultados e efectua-se uma análise dos mesmos, comparado-os com outras abordagens conhecidas e procurando justificar a razão de ser dos mesmos.

Finalmente, no **Capítulo 6**, são apresentadas as conclusões, onde são evidenciados os resultados obtidos, e o possível rumo a tomar num trabalho futuro.

Para além destes capítulos, a dissertação inclui ainda em anexo, num CD, gráficos e tabelas sumárias decorrentes da análise dos dados obtidos após a aplicação do programa às sequências de teste padrão, e o próprio programa, em código exposto de forma comentada.

Capítulo 2

Conceitos Biológicos e Computacionais

Antes de avançar, é necessário apresentar e esclarecer alguns dos conceitos de Biologia e de Biologia Computacional, que se encontram na base do trabalho realizado.

A biologia computacional é uma área de trabalho recente que compreende a utilização de técnicas computacionais e matemáticas para a resolução de problemas no domínio da Biologia, usualmente através da criação de programas de computador, modelos matemáticos ou ambos. O alinhamento de sequências, a previsão da estrutura tridimensional de proteínas, e as interacções proteína-proteína, são exemplos de problemas da biologia computacional.

A biologia computacional é também considerada a terceira e mais recente maneira de fazer experiências sobre organismos biológicos (ou baseadas em comportamentos biológicos). As duas maneiras anteriores são as experiências *in vivo* (efectuadas em organismos vivos) e as experiências *in vitro* (realizadas num ambiente artificial, em laboratório). A biologia computacional corresponde à maneira *in silico*, cuja denominação é baseada no nome dado aos processadores feitos em pastilhas de silício.

Envolvendo a biologia e a computação, interessa tornar claros os conceitos de cada uma das áreas presentes nesta dissertação.

2.1 Biologia Molecular

Toda a vida baseada no carbono depende da informação guardada no ADN (ácido desoxirribonucleico),[1] uma molécula de ácidos nucleicos que contém o conjunto de

[1]Muitas vezes, o ADN é conhecido pela sigla inglesa DNA, *Deoxyribonucleic Acid*.

instruções utilizadas para o desenvolvimento e funcionamento dos organismos vivos. Através de um processo complexo, o material constante no ADN de uma célula, o genótipo, vai dar origem a um indivíduo, cujos traços exteriormente observáveis são designados por fenótipo.

Em 1953, Watson e Crick decifraram o mistério da molécula de ADN, propondo o modelo de dupla hélice (ver Fig. 2.1), o que permitiu compreender melhor o mecanismo da expressão genética, que determina o que nós somos.

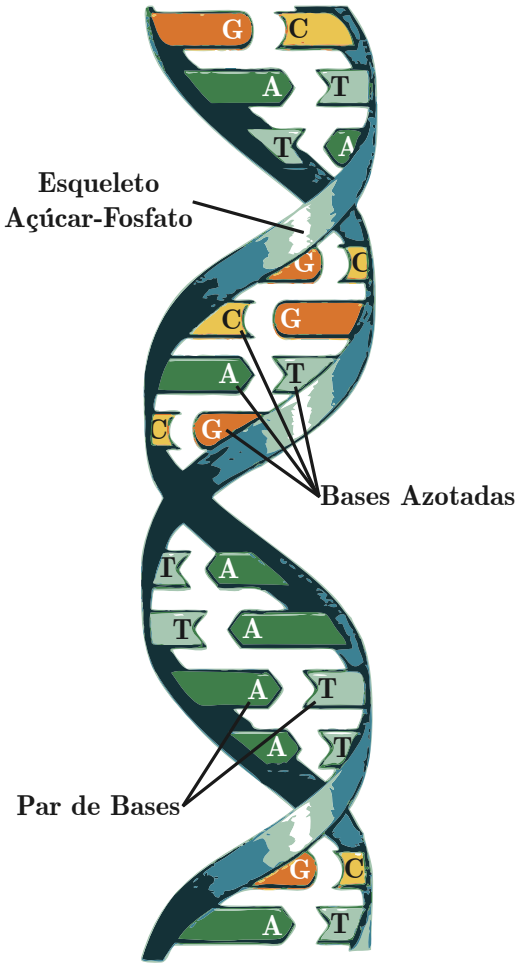


Figura 2.1: Molécula de ADN

A molécula de ADN tem codificada em si toda a informação hereditária de um organismo, o genoma. O genoma divide-se em segmentos denominados genes — o conjunto dos quais recebe o nome de genótipo — e ADN-lixo[2] (porções do genoma

[2]Referido, em inglês, por *"junk" DNA*.

para as quais não foi ainda identificada qualquer função). Os genes, eles próprios, são sequências de nucleótidos. Um nucleótido é formado por um fosfato, um açúcar e uma base. Existem quatro tipos de bases: Adenina *(A)*, Timina *(T)*, Guanina *(G)* e Citosina *(C)*. As bases ligam-se aos pares, sendo que A liga-se sempre com T, e G sempre com C, verificando-se o emparelhamento de bases complementares. Estas ligações são de tipo covalente, através de átomos de hidrogénio (ver Fig. 2.2).

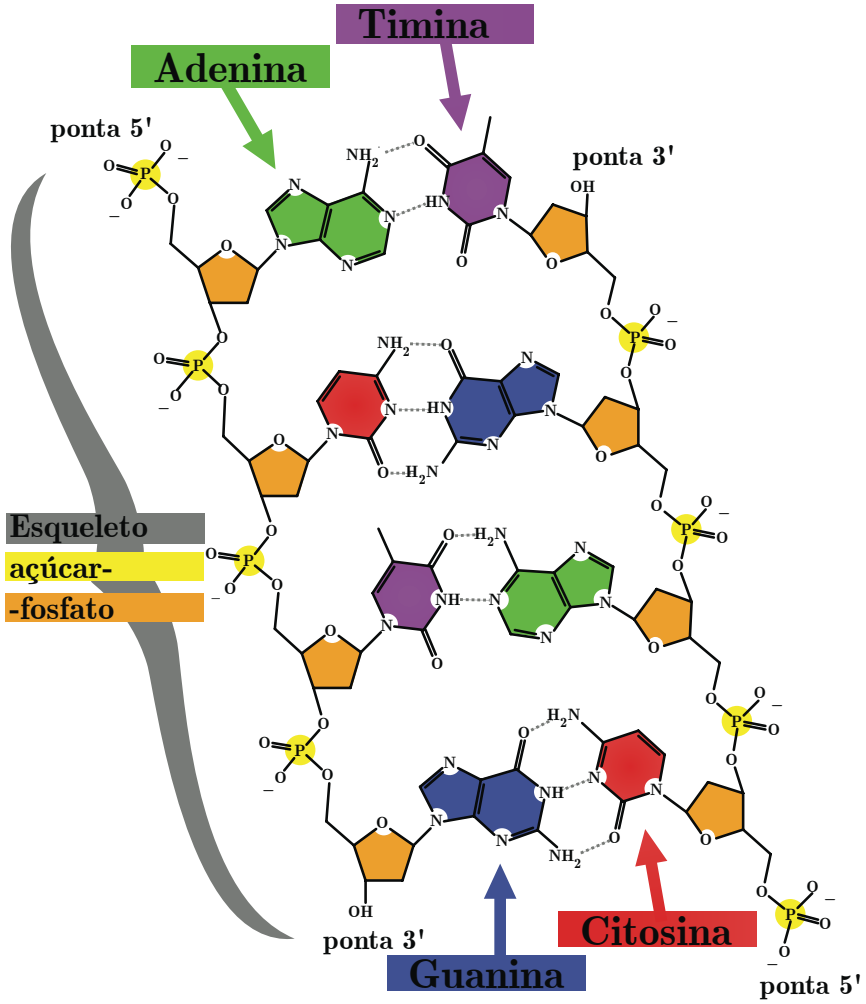


Figura 2.2: Esquema químico base da molécula de ADN com os nucleótidos

A título de exemplo, um ser humano, o *Homo Sapiens Sapiens*, possui cerca de 42.000 genes, divididos em 46 cromossomas (agrupamentos de genes, elementos reguladores e outras sequências de nucleótidos),[3] para definir todos os seus processos

[3]Sendo mais específico, as células somáticas (as células do corpo) possuem 46 cromossomas, ao

e características biológicas. As células humanas possuem 22 pares de cromossomas comuns a ambos os sexos (autossomas), mais dois cromossomas X para as mulheres ou um cromossoma X e outro Y para os homens (cromossomas sexuais), onde estão definidas as características sexuais. Também a título de exemplo, podem ser observados na Fig. 2.3 os nucleótidos do gene que codifica a proteína elastase pancreática humana (ELA1).[4]

```
   1 ttggtccaag caagaaggca gtggtctact ccatcggcaa catgctggtc ctttatggac
  61 acagcaccca ggaccttccg gaaaccaatg cccgcgtagt cggagggact gaggccggga
 121 ggaattcctg gccctctcag atttccctcc agtaccggtc tggaggttcc cggtatcaca
 181 cctgtggagg gacccttatc agacagaact gggtgatgac agctgctcac tgcgtggatt
 241 accagaagac tttccgcgtg gtggctggag accataacct gagccagaat gatggcactg
 301 agcagtacgt gagtgtgcag aagatcgtgg tgcatccata ctggaacagc gataacgtgg
 361 ctgccggcta tgacatcgcc ctgctgcgcc tggcccagag cgttaccctc aatagctatg
 421 tccagctggg tgttctgccc caggagggag ccatcctggc taacaacagt ccctgctaca
 481 tcacaggctg gggcaagacc aagaccaatg ggcagctggc ccagaccctg cagcaggctt
 541 acctgccctc tgtggactac gccatctgct ccagctcctc ctactggggc tccactgtga
 601 agaacaccat ggtgtgtgct ggtggagatg gagttcgctc tggatgccag ggtgactctg
 661 ggggcccccc ccattgcttg gtgaatggca agtattctgt ccatggagtg accagctttg
 721 tgtccagccg gggctgtaat gtctccagga agcctacagt cttcacccag gtctctgctt
 781 acatctcctg gataaataat gtcatcgcct ccaactgaac attttcctga gtccaacgac
 841 cttcccaaaa tggttcttag atctgcaata ggacttgcga tcaaaaagta aaacacattc
 901 tgaaagacta ttgagccatt gatagaaaag caaataaaac tagatataca tt
```

Figura 2.3: Gene que codifica a proteína elastase pancreática humana (ELA1)

As várias funções do ADN englobam o controlo da actividade celular (produzindo as características individuais e das espécies), a replicação (passando material genético de célula para célula e, por extensão, de geração para geração) e a permeabilidade a mutações (sofrendo alterações permanentes que serão passadas à descendência).

No entanto, não são os genes *per se* que permitem a realização dos processos biológicos necessários à vida. Os processos são despoletados (e processados) através das proteínas codificadas pelos genes. Assim, é de extrema importância conhecer em profundidade as proteínas que são responsáveis por todos os processos biológicos. Interessa assim, de início, conhecer a sequência de aminoácidos de uma proteína. Os aminoácidos são obtidos através da sequência de nucleótidos de um dado gene, sendo que cada conjunto de três nucleótidos, um codão, corresponde a um aminoácido.

passo que as células gaméticas (as células sexuais) possuem apenas metade, 23 cromossomas.

[4]Retirado do *"National Center for Biotechnology Information"*, em `http://www.ncbi.nlm.nih.gov/entrez/viewer.fcgi?db=nuccore&val=58331208`.

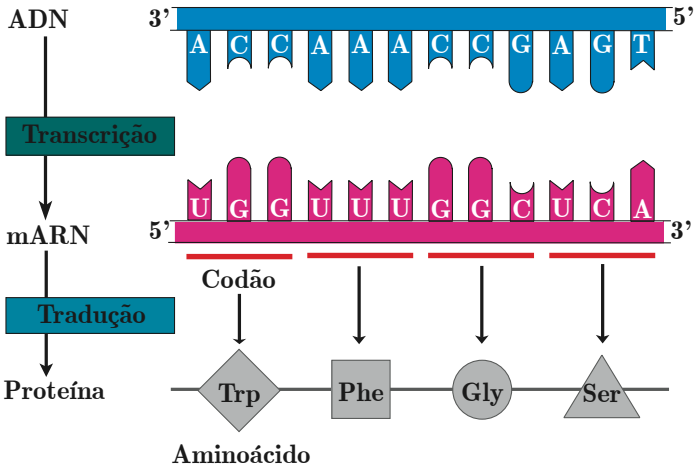


Figura 2.4: Sintetização dos aminoácidos de uma proteína

Na Fig. 2.4 pode ser observada esquematicamente o processo de expressão genética. Num primeiro passo, denominado transcrição, a molécula de ADN dá origem ao ARN (ácido ribonucleico).[5] O ARN é obtido transcrevendo uma das duas cadeias de nucleótidos de uma molécula de ADN — sendo a maior alteração a substituição do nucleótido Timina pelo o Uracil *(U)*. Num segundo passo, conhecido por tradução, cada codão dá origem a um aminoácido que comporá a proteína em causa. Diferentes codões podem codificar o mesmo aminoácido, já que os aminoácidos existentes são apenas 20 e as combinações possíveis de bases são 64.[6] A lista dos aminoácidos pode ser consultada nas Tabs. 2.1 e 2.2,[7] e na Fig. 2.8 pode ser observada a estrutura base de um aminoácido. Existem também codões que não codificam qualquer aminoácido mas, sim, pontos de quebra[8] nas proteínas.

Interessa referir que a relação entre os codões e os aminoácidos por eles codificados, mostrada na Tab. 2.1, é o chamado *código genético*. Dois aspectos chave do código genético são este ser redundante, já que vários codões podem codificar o mesmo

[5]O ARN é conhecido muitas vezes pela sigla inglesa RNA, *Ribonucleic Acid*. Existem vários subtipos, entre os quais, o ARN mensageiro *(mRNA)*, o ARN transportador *(tRNA)* e o ARN ribossómico *(rRNA)*.

[6]Dos quatro nucleótidos distintos existentes elevados às três posições disponíveis num codão, tem-se $4^3 = 64$.

[7]As denominações duplas, no formato de três letras, correspondem às variantes inglesa e portuguesa. Refira-se também que a última fila na Tab. 2.2 corresponde a casos em que não existe informação, ou esta é ambígua, quanto ao aminoácido em questão. "*Xaa*" corresponde a qualquer aminoácido, e "-" e "—" correspondem a falhas.

[8]Referidos como *stops*, em inglês.

Tabela 2.1: Lista dos aminoácidos e dos nucleótidos que os determinam

	U		**C**		**A**		**G**	
U	UU**U**	(**F**) Fenilalanina	UC**U**	(**S**) Serina	UA**U**	(**Y**) Tirosina	UG**U**	(**C**) Cisteína
	UU**C**	(**F**) Fenilalanina	UC**C**	(**S**) Serina	UA**C**	(**Y**) Tirosina	UG**C**	(**C**) Cisteína
	UU**A**	(**L**) Leucina	UC**A**	(**S**) Serina	UA**A**	*Stop*	UG**A**	(**C**) Cisteína
	UU**G**	(**L**) Leucina	UC**G**	(**S**) Serina	UA**G**	*Stop*	UG**G**	(**W**) Triptofano
C	CU**U**	(**L**) Leucina	CC**U**	(**P**) Prolina	CA**U**	(**H**) Histidina	CG**U**	(**R**) Arginina
	CU**C**	(**L**) Leucina	CC**C**	(**P**) Prolina	CA**C**	(**H**) Histidina	CG**C**	(**R**) Arginina
	CU**A**	(**L**) Leucina	CC**A**	(**P**) Prolina	CA**A**	(**H**) Glutamina	CG**A**	(**R**) Arginina
	CU**G**	(**L**) Leucina	CC**G**	(**P**) Prolina	CA**G**	(**Q**) Glutamina	CG**G**	(**R**) Arginina
A	AU**U**	(**I**) Isoleucina	AC**U**	(**T**) Treonina	AA**U**	(**N**) Asparagina	AG**U**	(**S**) Serina
	AU**C**	(**I**) Isoleucina	AC**C**	(**T**) Treonina	AA**C**	(**N**) Asparagina	AG**C**	(**S**) Serina
	AU**A**	(**I**) Isoleucina	AC**A**	(**T**) Treonina	AA**A**	(**K**) Lisina	AG**A**	(**R**) Arginina
	AU**G**	(**M**) Metionina	AC**G**	(**T**) Treonina	AA**G**	(**K**) Lisina	AG**G**	(**R**) Arginina
G	GU**U**	(**V**) Valina	GC**U**	(**A**) Alanina	GA**U**	(**D**) Aspartato	GG**U**	(**G**) Glicina
	GU**C**	(**V**) Valina	GC**C**	(**A**) Alanina	GA**C**	(**D**) Aspartato	GG**C**	(**G**) Glicina
	GU**A**	(**V**) Valina	GC**A**	(**A**) Alanina	GA**A**	(**E**) Glutamato	GG**A**	(**G**) Glicina
	GU**G**	(**V**) Valina	GC**G**	(**A**) Alanina	GA**G**	(**E**) Glutamato	GG**G**	(**G**) Glicina

aminoácido, e não ser ambíguo, uma vez que um codão codifica apenas um e só um aminoácido.

Tabela 2.2: Lista dos aminoácidos no formato de uma e três letras

1 L	**3 Letras**	**1 L**	**3 Letras**	**1 L**	**3 Letras**	**1 L**	**3 Letras**
A	Ala	R	Arg	N	Asn	D	Asp
C	Cys, Cis	Q	Gln	E	Glu	G	Gly, Gli
H	His	I	Ile	L	Leu	K	Lys, Lis
M	Met	F	Phe, Fen	P	Pro	S	Ser
T	Thr, The	W	Trp, Tri	Y	Tyr, Tir	V	Val
B	*Gln* ou *Glu*	*Z*	*Asn* ou *Asp*	*X*	*Xaa*	-	—

Ainda sobre os codões, interessa referir que existe um sentido de leitura dos nucleótidos, que permite determinar quais os codões existentes e, assim, os aminoácidos por eles codificados. Observando a Fig. 2.2, é possível ver as três partes constituintes de um nucleótido: a base (que define cada um dos nucléotidos e onde se faz a ligação entre as duas cadeias), a pentose (elemento comum a cada um dos nucleótidos) e o

fosfato (que permite a ligação entre nucleótidos adjacentes na mesma cadeia, ligando-se à pentose, e também elemento comum aos aminoácidos). A pentose (um açúcar) tem 5 pontas (ver Fig. 2.5), das quais duas, a 3' e a 5', permitem as ligações entre os nucleótidos. Cada uma das pontas 3' de um nucleótido liga-se com uma ponta 5' de outro nucleótido, por intermédio de um fosfato; isto em cada uma das cadeias da dupla hélice de ADN (uma num sentido e outra noutro), sendo este o sentido de leitura estabelecido.

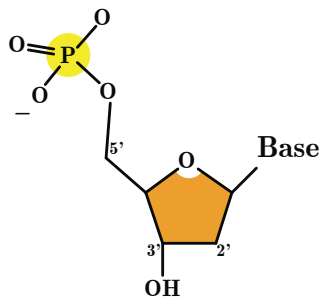


Figura 2.5: Pontas de um nucleótido

Regressando ao exemplo da proteína elastase pancreática humana, observe-se agora a sequência de aminoácidos dessa proteína, após a sua sintetização, na Fig. 2.6.

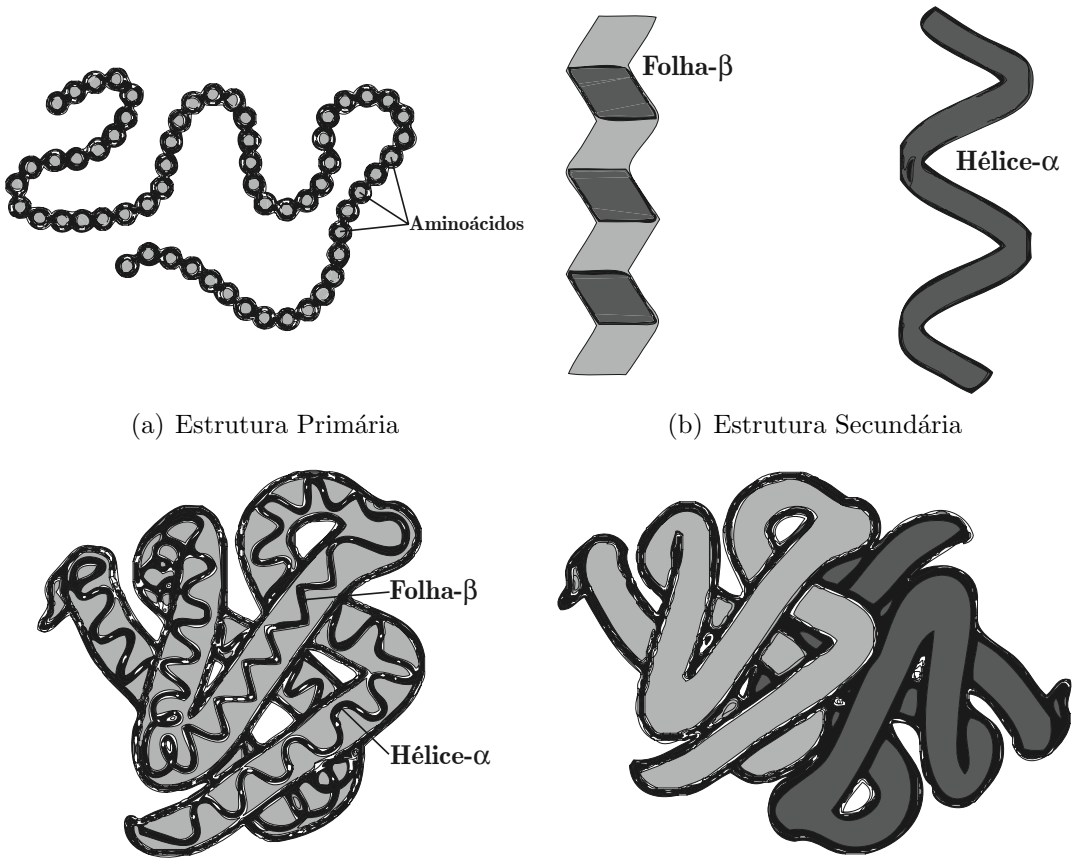
```
  1 MLVLYGHSTQ DLPETNARVV GGTEAGRNSW
 31 PSQISLQYRS GGSRYHTCGG TLIRQNWVMT
 61 AAHCVDYQKT FRVVAGDHNL SQNDGTEQYV
 91 SVQKIVVHPY WNSDNVAAGY DIALLRLAQS
121 VTLNSYVQLG VLPQEGAILA NNSPCYITGW
151 GKTKTNGQLA QTLQQAYLPS VDYAICSSSS
181 YWGSTVKNTM VCAGGDGVRS GCQGDSGGPL
211 HCLVNGKYSV HGVTSFVSSR GCNVSRKPTV
241 FTQVSAYISW INNVIASN
```

Figura 2.6: Aminoácidos da proteína elastase pancreática humana (ELA1)

Estruturas das Proteínas

As proteínas são estudadas em quatro níveis estruturais: a estrutura primária, definida apenas pela sequência de aminoácidos; a estrutura secundária, definida pelas

três formas básicas em que se agrupam os aminoácidos; a estrutura terciária, definida pela disposição espacial dos aminoácidos dentro de uma proteína (e que engloba as estruturas secundárias);[9] e a estrutura quaternária, que representa a agregação de várias cadeias de aminoácidos, como é o caso de algumas proteínas. Podem ser observados exemplos das várias estruturas na Fig. 2.7.



(a) Estrutura Primária (b) Estrutura Secundária

(c) Estrutura Terciária (d) Estrutura Quaternária

Figura 2.7: Diversas estruturas de uma proteína

A **estrutura primária** de uma proteína não é mais que a identificação da cadeia de aminoácidos da proteína. O conhecimento da estrutura primária de uma proteína é de extrema importância, tanto mais que é a partir desta que se pode chegar às restantes estruturas. Mas a estrutura primária também pode dar ideia da função da proteína, através da sua comparação com outras estruturas de proteínas já bem conhecidas. Assume-se que duas sequências com um grau de similaridade até à casa dos 50% terão uma função semelhante (e uma estrutura tridimensional semelhante) e

[9]Para uma primeira descrição das estruturas primárias, secundárias e terciárias, consulte-se, por exemplo, [Anf59], da autoria de Christian Anfinsen, um dos pioneiros da área da biologia molecular.

que descendem de um antecessor comum. Percentagens na casa dos 25 a 50% ainda apresentam um elevado grau de semelhança, com estruturas terciárias divergindo apenas em *voltas* na superfície da proteína. Esta percentagem relativamente baixa deve-se ao facto de haver nas proteínas pontos-chave que, se forem homólogos, indicam que a função das proteínas será, com uma probabilidade elevada, semelhante. É contudo necessário que esses pontos-chave sejam mantidos e bem conhecidos — i.e., que permitam uma conformação similar. Com valores acima dos 20% ainda é possível propor um modelo decente, se as proteínas pertencerem à mesma família. Percentagens inferiores a 20% já não oferecem certezas, podendo os resultados pender para ambos os lados. Tudo isto é possível devido a já ter sido provado que as estruturas terciárias de proteínas homólogas são mais invariantes que a sua estrutura primária [CN03]. É este o fundamento da Modelação Comparativa.

A **estrutura secundária** de uma proteína fornece informação quanto à forma que determinados segmentos de aminoácidos assumem quando se encontram em conjunto, através de ligações peptídicas — ligações de hidrogénio entre dois aminoácidos. Estas estruturas podem assumir uma de três formas: *hélices-*α, *folhas-*β e *voltas*.[10] São utilizadas também na representação tridimensional das proteínas, depois de conhecido o posicionamento espacial dos aminoácidos.

A **estrutura terciária** disponibiliza informação quanto à distribuição espacial dos aminoácidos de uma proteína — i.e., a forma que esta assume na sua conformação ou estrutura tridimensional. A estrutura tridimensional de uma proteína é de extrema importância, já que a sua função é influenciada em grande medida pela sua estrutura. Crê-se que, apesar de duas proteínas poderem ter estruturas primárias bastantes distintas, se a estrutura terciária for semelhante, possuem funções idênticas [CN03]. Há modelos de previsão da estrutura terciária que, de uma forma simples, tentam encaixar as formas definidas nas estruturas secundárias das proteínas, sendo que a estrutura terciária é também influenciada pela estrutura secundária de grupos de aminoácidos [SK03].

A **estrutura quaternária** está relacionada com a representação de proteínas que combinam mais que uma cadeia de aminoácidos, existindo também problemas na previsão da sua conformação. No entanto, é actualmente de importância inferior à da estrutura terciária, visto serem poucas as proteínas que possuem estruturas quaternárias. A relação existente entre as estruturas quaternária e terciária é, contudo, semelhante à relação existente entre as estruturas terciária e secundária.

[10] Do inglês α-*helices*, β-*sheets* e *loops*, respectivamente.

Conformação de Proteínas

De volta à estrutura terciária, a grande questão é conhecer de que maneira a proteína adquire a sua forma final, o que pode acontecer num espaço de tempo de poucos segundos — por vezes, apenas alguns milissegundos. Têm sido aventadas várias hipóteses sobre o processo de conformação de uma proteína (sendo a "hipótese termodinâmica", posposta por Anfinsen [Anf73], uma das mais conhecidas), mas nenhuma delas é isenta de falhas. Sabe-se já que a cadeia de aminoácidos define intrinsecamente a estrutura tridimensional da proteína que, por sua vez, define as funções dessa mesma proteína. Também já se conhecem vários tipos de factores que influenciam a conformação, nomeadamente forças electroestáticas, ligações de hidrogénio e interacções *van der Waals*, propensões intrínsecas e interacções hidrofóbicas, que influenciam a forma como os mesmos aminoácidos interagem [Dil90]. Actualmente o grande problema é descobrir uma função de avaliação (ou um algoritmo, dependendo da abordagem) que tenha em conta todos estes factores, de forma a fazer previsões mais precisas. Por outro lado, ainda há afinações que se podem fazer e, julga-se, descobrir mais alguns factores (ou melhorar os actuais) que influenciem a conformação da proteína no seu estado nativo. O estado nativo de uma proteína é a estrutura tridimensional que esta assume em função de interacções não-covalentes, tais como as interacções hidrofóbicas, interacções electro-estáticas e pontes de hidrogénio, à temperatura ambiente natural para essa proteína.

Apesar do acima exposto, é também sugerido que, em teoria, as forças responsáveis pela correcta conformação de uma proteína devem depender de princípios básicos de química e física, pensando-se que o conhecimento da sequência de aminoácidos deveria ser suficiente para especificar a estrutura tridimensional da proteína. Isto, em parte, porque a conformação *in vivo* é bastante mais célere que a conformação *in vitro*. Com todas estas condicionantes e contradições, este problema continua a revelar-se bastante difícil, contando já com perto de 50 anos de investigação, e continua a ser ainda referido como o "problema da conformação de proteínas" [Ric91].

Assim, a previsão da conformação de proteínas envolve a descoberta da relação entre os genes, através dos quais se obtém a sequência de aminoácidos de uma proteína, e o que as proteínas fazem (a sua função), baseadas na sua estrutura.

Na Fig. 2.8 podem ser observadas as ligações entre os vários elementos de um aminoácido, bem como a estrutura base cada um dos aminoácidos: $NH_2 + \mathbf{R} + COOH$ — i.e., um *grupo amino*, um ***grupo R*** e um *grupo carboxil*. Note-se que é o grupo *R*, o elemento variável entre aminoácidos, que define qual o aminoácido em questão.

Grupo R é um nome genérico, em aminoácidos, para uma cadeia lateral — parte de uma molécula que está ligada a uma estrutura principal —, cujo nome deriva de "radical".

As ligações entre aminoácidos são feitas entre as pontas (as moléculas) NH_2 e COOH (os grupos amino e carboxil), gerando ligações peptídicas $CO-NH$, como pode ser observado nas Figs. 2.9(a) e 2.9(b). A leitura de uma sequência de uma proteína faz-se da ponta NH_2, 5', para a ponta COOH, 3', como já foi referido anteriormente (ver Fig. 2.2).

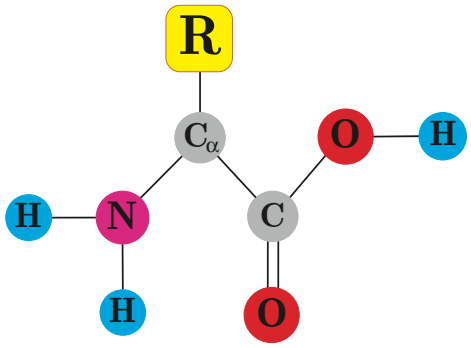


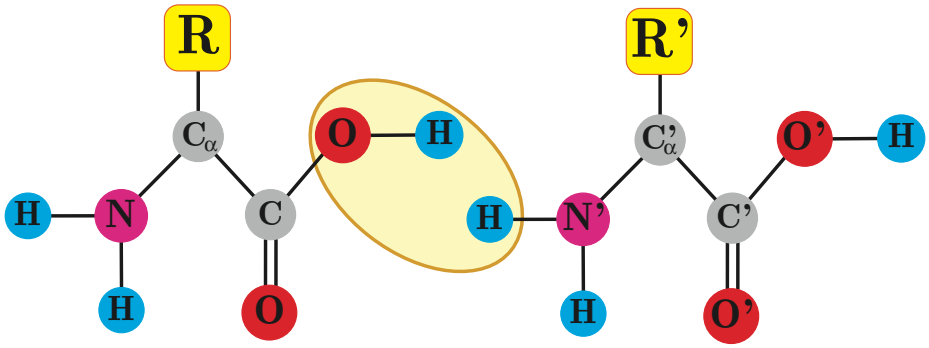
Figura 2.8: Estrutura de um aminoácido

A sucessão linear dos aminoácidos define determinada proteína. No entanto, as propriedades biológicas de uma proteína não resultam da cadeia de aminoácidos como um objecto linear, mas da forma tridimensional bastante compacta que esta fita de aminoácidos assume no seu ambiente natural. A forma tridimensional de uma molécula da proteína é definida pelos tipos dos aminoácidos, se estes tem propriedades hidrofóbicas ou hidrofílicas que os afastam ou aproximam, respectivamente, da superfície da proteína. A estrutura da proteína também é afectada por cargas eléctricas dos aminoácidos e pela capacidade que aqueles têm de interagir com os seus vizinhos [CN03].

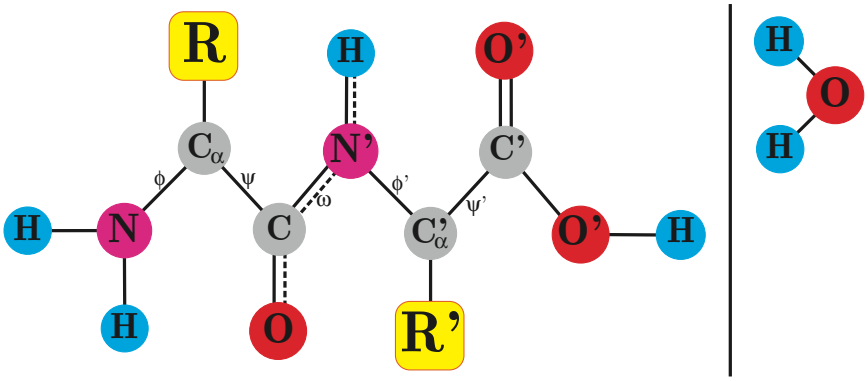
Nas proteínas, bem como nas outras moléculas, existe uma série átomos ligados de forma covalente, de modo a permitir a cadeia contínua da molécula, a que é dado o nome de esqueleto.[11] Este esqueleto é muitas vezes utilizado para uma representação simplificada da proteína, denotando apenas a estruturas básicas das proteínas, nomeadamente as formas representadas na estrutura secundária, já anteriormente mencionada. À partida, conhecendo o esqueleto, é depois tudo "apenas"

[11] Em inglês, o termo usado é *backbone*.

uma questão de adicionar as cadeias laterais (os *grupos R*) para se obter a estrutura completa da proteína.



(a) Ligação entre aminoácidos

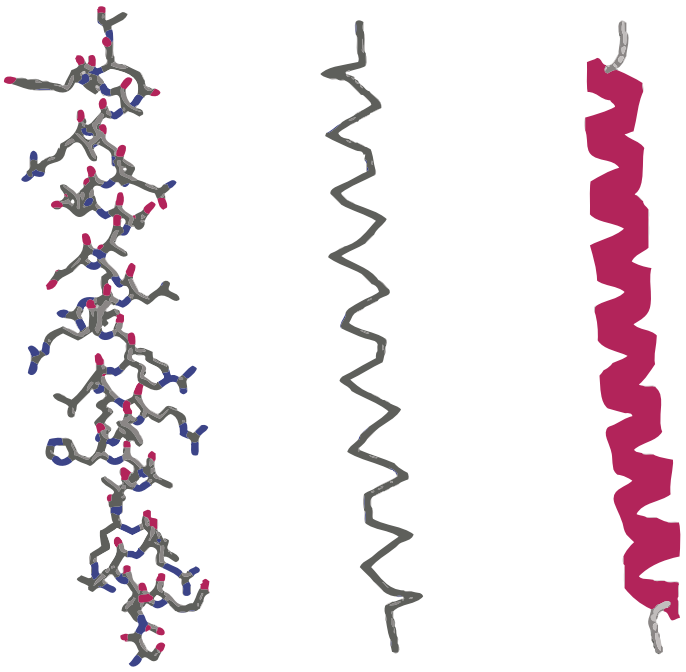


(b) Ângulos diedros na ligação

Figura 2.9: Ligações entre aminoácidos

Note-se, também, na Fig. 2.9(b) a existência dos ângulos diedros ω, ϕ e ψ e também do carbono α (C_α). Nas ligações peptídicas, o C_α define o centro de rotação dos aminoácidos relativamente aos outros a que se ligam, sendo as rotações que, posteriormente, darão origem às estruturas secundárias e terciárias de uma proteína, posicionando espacialmente cada um dos aminoácidos relativamente aos outros. Os modelos mais complexos de previsão da conformação de proteínas trabalham com estes ângulos para a determinação do esqueleto de uma proteína, preenchendo-o depois com as cadeias laterais dos aminoácidos para obtenção do modelo completo, como pode ser observado na Fig. 2.10, onde se encontram três representações possíveis para

uma *hélice-α*. Ainda sobre as ligações peptídicas entre aminoácidos, note-se que dois átomos de hidrogénio e um de oxigénio são afastados, dando origem a uma molécula de água (H_2O). Aos aminoácidos, depois da ligação e da libertação da molécula de água, dá-se o nome de resíduos.[12] Quando dois aminoácidos se ligam, um deles sofre uma rotação próxima dos 180°, representada pelo ângulo ω. Os outros dois ângulos têm mais graus de liberdade, sendo eles essencialmente os responsáveis pela estrutura da proteína, ao posicionarem os resíduos e também, dessa forma, as cadeias laterais.[13]



(a) Esferas (b) Esqueleto (c) Fita

Figura 2.10: Diversas formas de representar uma *hélice-α*

2.2 Computação Evolucionária

A inspiração biológica da computação evolucionária surge da observação da evolução dos seres vivos, que favorece a sobrevivência dos mais aptos, tal como foi afirmado por Charles Darwin. A teoria proposta por Darwin resultou de muitas observações

[12] Apesar de aminoácidos e resíduos não serem estritamente a mesma coisa, utilizar-se-á, deste ponto em diante, indistintamente o termo aminoácido, no contexto das proteínas, por uma questão de facilidade de interpretação do texto.

[13] A título informativo, também as cadeias laterais têm conformações precisas, mas este tema está fora do âmbito desta dissertação. Para mais informação consulte-se, por exemplo, [Sil99].

feitas ao longo da viagem que fez no navio Beagle, em particular observando que no Arquipélago das Galápagos havia espécies de tentilhões, com origens comuns a espécies do continente sul-americano, que tinham sobrevivido por se terem conseguido adaptar ao meio ambiente (distinto do seu *habitat* original) existente nas ilhas, tendo maiores hipóteses de passarem as suas características à geração seguinte [Dar05]. Note-se que este processo de adaptação acontece por acaso e não intencionalmente — daí que em cada geração, para além dos indivíduos considerados normais, surjam esporadicamente alguns indivíduos que estão condenados ao fracasso e outros com maiores probabilidades de sucesso. Esta diferenciação, muitas vezes, só é notada com a acumulação destas novas características, que eventualmente acabam por dar origem a uma nova espécie.[14]

Outro trabalho que também deu um grande contributo para as teorias evolutivas foi o de Gregor Mendel. Foi Mendel que constatou, através de experiências com ervilhas, que os indivíduos de uma nova população resultam do cruzamento das características dos seus progenitores [Men96]. Mendel, para além do cruzamento (também conhecido por recombinação), também se apercebeu de como as características presentes nos indivíduos são afectadas por mutações aleatórias. Ao conjunto das teorias de Darwin e Mendel, dá-se o nome de *neodarwinismo*, onde se conjuga o processo de selecção natural com os diversos mecanismos de variação (mutação e recombinação).

A área da computação evolucionária baseia-se nos princípios *neodarwinistas*, aplicando-os computacionalmente à modelação de um problema complexo. Existem diferentes variantes, nomeadamente os Algoritmos Genéticos que fazem uso de mecanismos de selecção e dos operadores genéticos. Também fazem uso da terminologia biológica, utilizando, por exemplo, termos como cromossomas, genes e alelos (os diversos valores que um gene pode assumir) para representarem, respectivamente, indivíduos, variáveis e valores.

A título de curiosidade, o Homem tem tido um papel activo na selecção de indivíduos e na utilização dos operadores genéticos. O exemplo máximo desse facto é a domesticação, quer de animais, quer de plantas. Por exemplo, no caso dos cães, o Homem preferiria sempre como animal de companhia, ou de caça, aquele que fosse mais afável, ou melhor caçador, favorecendo a selecção *natural* nesse sentido. O Homem procuraria também cruzar indivíduos de raças diferentes de maneira a obter um novo indivíduo com características de ambos os progenitores — e.g., um melhor faro

[14] Apesar de não cobrir todos os casos possíveis, uma definição comum de espécie é um grupo de organismos capazes de se cruzarem entre si e gerarem uma descendência fértil.

combinado com uma maior resistência física. Alguns dos animais e plantas domesticados (especialmente os de quinta) estão de tal forma dependentes do Homem — i.e., adaptados às condições de sobrevivência proporcionadas pelo Homem —, que se crê que se o Homem desaparecesse, muitas dessas espécies desapareceriam também.

Algoritmos Genéticos

Os Algoritmos Genéticos são métodos de procura estocástica inspirados no princípio *neodarwinista* da selecção natural e de variação. Foram inicialmente propostos por John Holland [Hol92], e têm vindo a ser aplicados com sucesso a diferentes tipos de problemas, para os quais os métodos exactos se revelam ineficazes.

A sua escolha, para o trabalho desenvolvido para a dissertação, prendeu-se com, para além de aspectos intrínsecos ao problema em estudo (um típico caso de optimização), a inspiração que foram beber à evolução natural dos seres vivos, usando-se, assim, uma técnica inspirada no mundo natural para resolver um problema também do mundo natural.

Os algoritmos genéticos são tipicamente implementados como uma simulação de computador na qual uma população de representações abstractas de soluções candidatas (chamadas de cromossomas ou indivíduos) são evoluídas para resolver problemas complexos. Tradicionalmente, as soluções são representadas como uma cadeia de zeros e uns, mas também é possível usar outras representações, melhor adaptadas ao problema em questão. A evolução começa com uma população de indivíduos gerados aleatoriamente. Em cada geração, vários indivíduos são escolhidos estocasticamente da população actual, modificados — i.e, mutados e/ou recombinados —, avaliados e, de entre os modificados, alguns formam uma nova população, que se torna a população actual na próxima iteração do algoritmo. Pode ser observado na Fig. 2.11 (adaptada de [HYTY05]) o esquema básico de um algoritmo genético.

2.3 Biologia Molecular e Computação Evolucionária

O recurso a algoritmos genéticos para resolver o problema da Previsão da Conformação de Proteínas — e mais especificamente na Previsão *ab initio* — permite "fechar o círculo", com a utilização de algoritmos de inspiração biológica para resolver problemas relacionadas com biologia molecular.

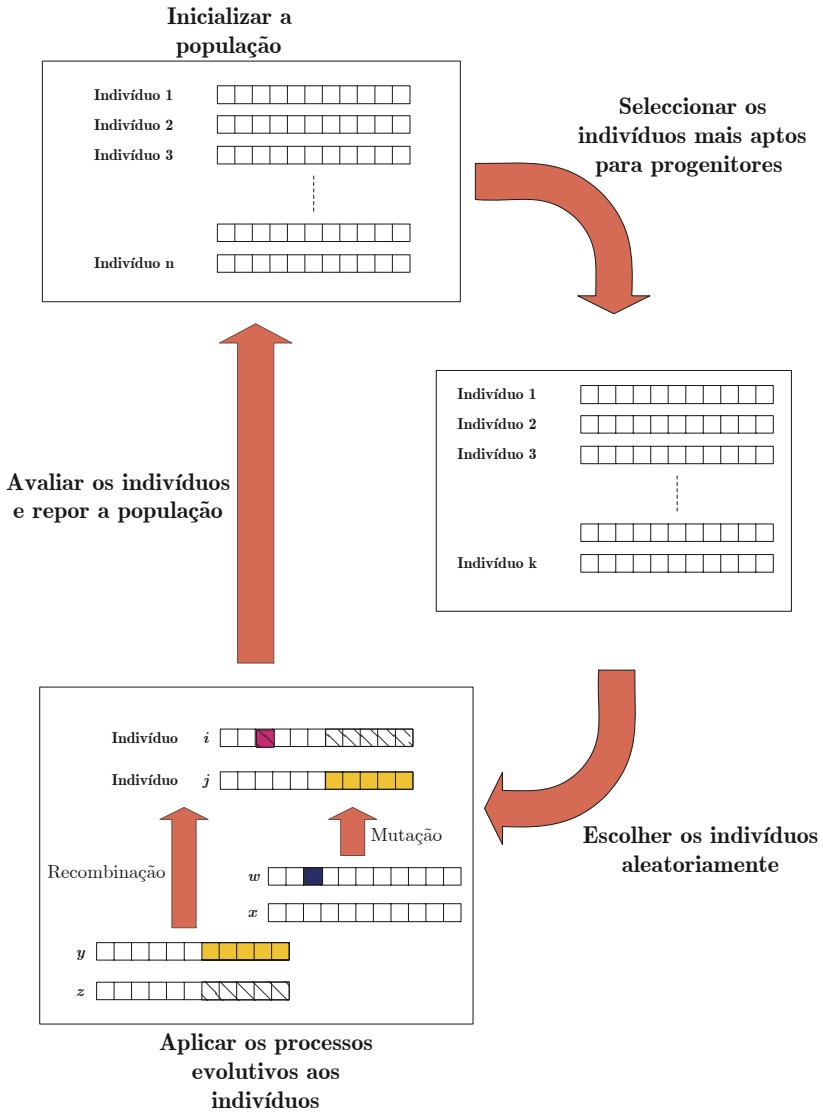


Figura 2.11: Esquema de um algoritmo genético clássico

A computação evolucionária (e em especial, os algoritmos genéticos) é também, até ao momento, uma das melhores abordagens ao problema da previsão da conformação de proteínas. Isto é, os algoritmos genéticos são alvo da preferência da investigação nesta área devido a vários factores, tais como:

- a previsão da conformação de proteínas ser um problema NP-difícil [BL98];
- o facto de não se conhecer*a priori*, com exactidão, a melhor solução;
- o facto de esta ser das poucas técnicas computacionalmente comportáveis para espaços de soluções tão vastos, como é o caso das conformações de proteínas, ao mesmo tempo que apresentam resultados suficientemente expressivos.

A título de exemplo, uma proteína com 100 aminoácidos tem cerca de 10^{130} possibilidades de conformação [WD83]. Uma outra ideia da complexidade envolvida neste problema pode ser obtida através da observação da Fig. 2.12, onde são apresentadas as várias condicionantes a ter em linha de conta no conhecimento da função de uma proteína, desde a sequência (a fórmula química) até à sua funcionalidade complexa (devido à sua estrutura nativa).

sequência → superfície potencial → formas dinâmicas →
→ formação da estrutura → **complexidade**

Figura 2.12: Complexidade na descoberta da função de uma proteína

No capítulo seguinte passa-se a descrever o estado da arte da aplicação da computação evolucionária à biologia molecular, no domínio da conformação de proteínas.

Capítulo 3

Estado da Arte

Neste capítulo é apresentado o Modelo HP, o modelo escolhido e estudado no âmbito desta dissertação, bem como as abordagens mais conhecidas a este modelo. Sendo uma representação de um problema bastante mais complexo, o Modelo HP é estudado essencialmente como base de experiências e em provas de conceito da aplicação de técnicas informáticas *ab initio* à previsão de conformação de proteínas, que depois poderão ser aplicadas a modelos mais precisos.

3.1 Modelos para a Conformação de Proteínas

Dentro do domínio da previsão da conformação de proteínas com recurso a técnicas evolucionárias, existem vários modelos simplificados, entre os quais se destacam o "*Protein Structure Prediction*" (PSP),[1] o "*Lattice Polymer Embedding*" (LPE), o "*Charged Graph Embedding*" (CGE) e o seleccionado Modelo Hidofóbico-Polar (Modelo HP).

De uma forma ou de outra, qualquer um destes modelos faz simplificações à realidade da conformação de proteínas, focando essencialmente o posicionamento dos aminoácidos e prestando posteriormente apenas atenção a algumas das características dos aminoácidos que influenciam a conformação das proteínas, como é o caso da hidrofobicidade ou da carga. A excepção é o modelo PSP, que é mais fiel à realidade ou, pelo menos, capta mais factores envolvidos na conformação.

Todos estes, à excepção do modelo PSP, são modelos discretos que podem ser considerados interessantes para a área da computação, mas que, por outro lado, podem

[1] Houve a opção de se usarem aqui as denominações originais em inglês, para este modelo e para os dois seguintes, devido a serem aquelas pelas quais eles são habitualmente referidos. Uma tradução possível para os nomes dos três modelos é: PSP — Previsão da Estrutura de Proteínas; LPE — Malha com Embutimento de Polímeros; CGE — Embutimento de Grafos de Cargas.

também ser considerados menos satisfatórios para biólogos. Contudo, representam uma boa solução de compromisso, uma vez que, como foi já referido anteriormente, é bastante mais barata a realização de experiências com recurso a modelos informáticos do que a realização de experiências laboratoriais *in vitro*. Em [CDK03] podem ser encontradas várias vantagens e desvantagens da utilização destes modelos, as quais passam a ser enunciadas de seguida, começando-se pelas desvantagens:

- perde-se a granularidade do problema original, principalmente ao nível das ligações entre aminoácidos (quer em termos de distâncias entre os aminoácidos, quer em termos de ângulos das ligações entre os aminoácidos);

- pormenores da estrutura da proteína, tais como as energias de ligação entre aminoácidos ou as cargas desses mesmos aminoácidos, não são representados nestes modelos, ou são-no de forma não muito precisa.

Já no campo das vantagens, destacam-se as seguintes, que justificam a utilização destes modelos, principalmente do ponto de vista da engenharia:

- os modelos em malha permitem a simulação de um número bastante vasto de mudanças ao nível da conformação — o que não aconteceria com modelos mais minuciosos, que precisariam de muito mais tempo para explorar um número idêntico de mudanças;

- a simulação de modelos atómicos (em vez da simulação à escala molecular dos aminoácidos) envolvem tantos parâmetros e aproximações, que a sua validade seria tão duvidosa quanto a de modelos mais simples;

- os modelos em malha podem ser utilizados para a pesquisa exaustiva de determinadas conformações, ou outros aspectos em estudo (ao contrário de modelos mais pormenorizados), que depois podem ser utilizados para a dedução de propriedades estatísticas de conformações.

Seguem-se breves descrições dos modelos mencionados, à excepção do modelo HP, que, devido ao seu papel nesta dissertação, será estudado em detalhe mais adiante na *Secção 3.2*.

Protein Structure Prediction

O modelo PSP é um modelo generalista, não discreto, em que uma proteína é definida:

- pela lista de todos os átomos das moléculas que a constituem (entre essa moléculas, os aminoácidos);
- pelas ligações entre moléculas, pelos comprimentos e ângulos das ligações entre as moléculas;
- pelas forças existentes entre aminoácidos (local e globalmente), que se encontram dispostos num espaço tridimensional.

Neste modelo, a função de avaliação toma em conta a soma de todos estes elementos (não se limitando às relações locais entre átomos, mas também tendo em conta as relações globais), procurando encontrar o mínimo global da energia de conformação.

Este foi um modelo proposto por Ngo e Marks [NM92], com o intuito, entre outros, de provar que o problema da previsão da conformação de proteínas é um problema NP-difícil. Apesar de ser um modelo mais próximo do real que os modelos discretos (onde se inclui o modelo HP), acabou por ser preterido, devido à sua complexidade. Contudo, há abordagens complexas que o utilizam, como é o caso do projecto coordenado por M. Karplus na Universidade de Harvard, que dá pelo nome de CHARMM, de "*Chemistry at HARvard Macromolecular Mechanics*", e que inclui funções de minimização de energia, campos de forças, dinâmica de moléculas e simulações de Monte Carlo [GS03]. Este projecto tem apresentado alguns resultados interessantes, mas a sua utilização (desde a configuração à interpretação dos resultados obtidos) é, em muito, dependente da experiência de quem o está a utilizar, podendo levar a resultados ambíguos [Sch06].

Lattice Polymer Embedding

O modelo LPE faz uso de um reticulado (ou malha) tridimensional finito, onde cada aminoácido ocupa uma posição. A cadeia de aminoácidos tem de ser contínua e não pode existir uma posição que seja ocupada por mais que um aminoácido. O objectivo deste modelo é apresentar uma conformação que minimize a energia necessária para dispor os aminoácidos numa fita contínua que seja o mais compacta possível. Contudo, este modelo não adopta quaisquer das propriedades dos aminoácidos, dando apenas importância à disposição de cada um deles e à forma que adoptam no conjunto, o que

não torna este modelo, de forma alguma, realista. Mesmo assim, é uma base possível para outros modelos discretos que usem a mesma forma de dispor os aminoácidos e acrescentem as respectivas propriedades (ou parte delas).

Charged Graph Embedding

O modelo CGE, tal como o modelo LPE, também dispõe os aminoácidos como se tratassem de contas numa malha tridimensional. Contudo, permite que os caminhos entre aminoácidos se cruzem, e toma em conta a carga dos aminoácidos. A função de avaliação toma em conta a distância entre cada par de aminoácidos adjacentes e também as suas cargas: positivas, neutras ou negativas. Apesar de captar uma das propriedades dos aminoácidos que, de facto, tem uma quota parte de importância na conformação das proteínas, o facto de permitir o cruzamento de caminhos entre aminoácidos leva a que o modelo seja considerado pouco realista.

3.2 Modelo HP

O Modelo Hidrofóbico-Polar (Modelo HP), proposto por K. Lau e K. Dill [LD89], pretende modelar, de uma forma simples, a conformação de proteínas. Neste modelo são postas de parte algumas das propriedades dos aminoácidos e das ligações entre estes e, assim, os 20 aminoácidos existentes são divididos em apenas dois tipos: hidrofóbicos *(H)* e polares *(P)*, de acordo com esta propriedade existente nos aminoácidos (ver Tab. 3.1, baseada em [HYTY05]). De resto, este modelo partilha características com os modelos LPE e CGD, dispondo os aminoácidos como fossem contas numa malha (reticulado) bidimensional ou tridimensional — no Modelo HP 2D ou 3D, respectivamente.

Como foi já referido no *Capítulo 1*, os aminoácidos hidrofóbicos procuram estar no interior das proteínas, afastando-se das soluções aquosas onde estas se encontram; os polares encontram-se no exterior das proteínas, em contacto com as soluções aquosas. Ao serem respeitadas estas duas propriedades, as proteínas tendem a ser moléculas bastante densas, com um núcleo fortemente hidrofóbico (devido à sua constituição por aminoácidos hidrofóbicos) rodeado de aminoácidos polares. Note-se que não é permitida a sobreposição de aminoácidos.

O modelo HP, com base nas propriedades hidrofóbicas e polares dos aminoácidos, engloba apenas as interacções atractivas entre os aminoácidos, ficando de fora, por

Tabela 3.1: Propriedades hidrofóbicas e polares dos aminoácidos

Aminoácido	**Hidrofobicidade**
Alanina	H
Cisteína	P
Aspartato	P
Glutamato	P
Fenilalanina	H
Glicina	H *ou* P
Histidina	P
Isoleucina	H
Lisina	P
Leucina	H
Metionina	H
Asparagina	P
Prolina	H
Glutamina	P
Arginina	P
Serina	P
Treonina	P
Valina	H
Triptofano	H
Tirosina	P

exemplo, as interacções repulsivas, que é necessário representar em modelos mais próximos do real. Contudo, sabe-se que as interacções atractivas entre aminoácidos são aquelas que mais afectam a conformação das proteínas. Um exemplo deste modelo pode ser observado na Fig. 3.1,[2] onde os aminoácidos hidrofóbicos são representados pelos quadrados escuros, e os aminoácidos polares pelos quadrados claros.

Este modelo, na sua vertente bidimensional, acabou por ser o objecto de estudo da dissertação por várias razões, das quais se destacam o facto de ser um modelo bastante documentado e referido na literatura da área — bastante mais que os restantes modelos considerados — e também o facto de, apesar de ser um modelo simples, captar alguns dos aspectos chaves na conformação de proteínas, nomeadamente a hidrofobicidade ou polaridade dos aminoácidos.

No modelo HP 2D, é norma que a sequência de aminoácidos de uma proteína seja

[2]A leitura da sequência deve ser iniciada seguindo-se as setas apresentadas na figura, apostas ao enrolamento.

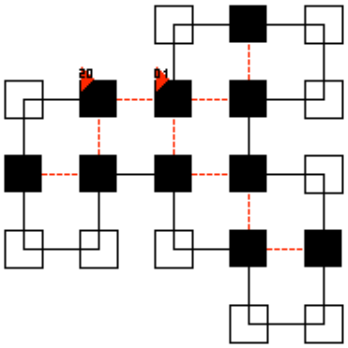


Figura 3.1: Modelo HP para a sequência HPHPPHHPHPPHPHHPPHPH

representada por um vector $s = (s^{(1)}, \ldots, s^{(n)})$, com $s^{(i)} \in \{H,P\}$, sendo n o número de aminoácidos da proteína. No caso do exemplo apresentado na Fig. 3.1, tem-se $s = (H,P,H,P,P,H,H,P,H,P,P,H,P,H,H,P,P,H,P,H)$.

Já a representação da localização de cada aminoácido pode variar consoante a abordagem, como se poderá observar em secções posteriores, sendo uma de três possíveis [KHSP99]:

1. *Matriz de distâncias*, onde a posição de cada aminoácido é obtida através de uma matriz de distâncias;

2. *Coordenadas cartesianas*, onde cada aminoácido é posicionado usando o sistema de eixos cartesianos, não dependendo da posição de outros aminoácidos;

3. *Coordenadas internas*, dependendo a posição de um aminoácido da posição do aminoácido que o precede na sequência. Este método tem duas variantes:

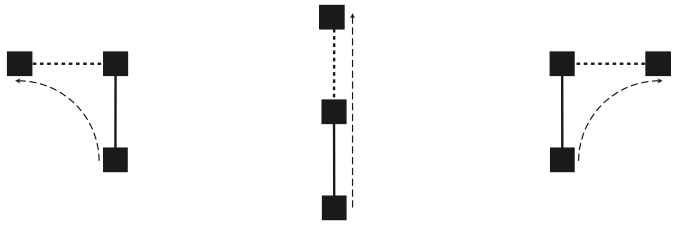
   (a) *direcções absolutas*, onde as direcções que definem a posição do aminoácido são imutáveis e com origem no aminoácido precedente;

   (b) *direcções relativas*, onde as direcções são obtidas em função da direcção que posicionou o aminoácido anterior.

A representação mais comum é a feita com coordenadas internas relativas, a qual pode ser observada na Fig. 3.2. Esta escolha prende-se com o facto de a codificação do posicionamento dos aminoácidos com coordenadas internas ter um tamanho menor e ser mais simples que a codificação das restantes representações; e também pelo facto

de, quando comparada com a representação com coordenadas internas absolutas, esta representação evitar um dos casos mais óbvios de sobreposição de aminoácidos, quando, ao posicionar um aminoácido, se "recua" e posiciona-se este em cima do anterior.

Assim, regra geral, cada conformação de uma proteína é representada por um vector $c = (c^{(1)}, \ldots, c^{(n-1)})$, com $c^{(i)} \in \{L,F,R\}$ e sendo n o número de aminoácidos, onde cada valor designa um ângulo de torção (esquerda, frente ou direita, respectivamente) do aminoácido actual relativamente ao anterior.[3] Habitualmente, o vector para representação da conformação é menor em uma unidade que o vector da sequência. Tal deve-se ao facto de uma dada direcção estar compreendida entre quaisquer dois aminoácidos, logo havendo mais um aminoácido que as direcções entre eles.



(a) Esquerda (b) Frente (c) Direita

Figura 3.2: Representação das várias direcções numa conformação

Interessa também ressalvar que em algumas abordagens, o vector c é menor em duas unidades que o vector s — i.e., tem-se $c = (c^{(1)}, \ldots, c^{(n-2)})$, com $c^{(i)} \in \{L,F,R\}$. Esta diferença de representação deve-se a ser também comum assumir-se que a primeira direcção é constante, não sendo, assim, necessário representá-la, dado que conformações que apenas difiram por rotação relativamente à origem são, para todos os efeitos, idênticas.

Portanto, tendo-se em conta o exposto acima, relativamente ao posicionamento dos aminoácidos, tem-se que para se obter conformação apresentada na Fig. 3.1 é necessária sequência de direcções $c = (F,R,F,R,R,L,L,R,F,R,R,L,R,L,L,R,R,F,R)$.[4]

No modelo HP, as várias soluções são avaliadas em função da energia despendida para obter a conformação em questão. As proteínas procuram uma energia mínima

[3]Refere-se aqui a notação inglesa mais comum com L, F e R (*left*, *forward* e *right*) para esquerda, frente e direita.

[4]Note-se que a primeira direcção, F, se for considerada fixa, poderá não ser representada na sequência necessária para obter a conformação.

de conformação para assumirem o seu estado nativo, enrolando-se no reticulado quadrangular, posicionando cada aminoácido 90° à esquerda, 90° à direita ou à frente do aminoácido precedente, e procurando juntar os aminoácidos com propriedades hidrofóbicas, como foi já referido anteriormente. A conformação do estado nativo procura maximizar os contactos H–H entre aminoácidos não adjacentes e, em caso algum, pode ter aminoácidos em posições sobrepostas. Os contactos H–H podem ser identificados na Fig. 3.1 pelas linhas a tracejado, que são nove, podendo-se observar, então, que a energia mínima de conformação é de -9 unidades, tendo-se assumido que cada contacto H–H vale -1 ponto e os outros 0 pontos.

Dito de outra forma, para dois aminoácidos não adjacentes, tem-se que $E_{HH} = -1$ e $E_{HP} = E_{PP} = 0$, e também que, sendo l o número de ligações (contactos) H–H não adjacentes, a energia mínima é dada por $E(l) = -l$. Assim, quanto menor for a pontuação obtida (energia despendida) em determinada conformação, melhor será a qualidade da solução.

Para além desta forma de avaliação das soluções, o modelo HP originalmente propunha também um outro mecanismo para comparar soluções (com a mesma energia). A distância entre duas conformações era dada por $d(c_1,c_2) = min[S(c_1-c_2),S(c_1+c_2)]$, sendo c o vector com as direcções e S a soma dos valores absolutos de cada uma das direcções (-1, 0 ou 1 — L, F e R, respectivamente). Contudo, esse mecanismo não procura determinar se uma é melhor que a outra, mas sim verificar a distância entre soluções. No entanto, na prática, este mecanismo é ignorado ou substituído por outros — mas continua a ser utilizado (este ou uma variação), por exemplo, em abordagens com fundos de progenitores,[5] para a selecção de indivíduos para recombinação, baseados na semelhança ou dissemelhança dos mesmos.

Resumindo, o problema do modelo HP pode ser formulado como: dada uma sequência s de aminoácidos, procura-se obter uma conformação c^* de s — i.e., $c^* \in C(s)$ — tal que $E(c^*) = min\{E(c) \mid c \in C(s)\}$, onde $C(s)$ é o conjunto de todas as conformações válidas para s.

Opção pelo Modelo HP 2D

Nesta dissertação, o modelo HP estudado foi o bidimensional (2D). Esta opção pelo modelo 2D deveu-se a duas importantes vantagens do modelo bidimensional sobre o

[5]Fundo de progenitores é uma tradução possível para a expressão inglesa *"parents' pool"*, que quer significar um subconjunto de indivíduos (à partida, mais qualificados) que serão a base para a selecção dos progenitores.

modelo tridimensional [LD89]:

- o *ratio* superfície/volume (um dos principais determinantes do comportamento físico de proteínas) de longas cadeias tridimensionais é semelhante ao *ratio* superfície/volume de cadeias bidimensionais mais pequenas;
- para um dado tamanho de uma cadeia, o esforço computacional é menos exigente no modelo bidimensional — sendo este, dos dois factores, o mais preponderante.

Por outro lado, a principal simplificação de um modelo bidimensional relativamente a um tridimensional consiste em ignorar a *profundidade*, o que leva a que haja menos conformações entre pares de aminoácidos (quatro em vez de seis possíveis) e, consequentemente, menos vizinhos para ligações. Também, por outro lado, a falta da terceira dimensão torna as conformações possíveis no modelo bidimensional mais restritas.

Contudo, e apesar destas diferenças, tal como é indicado em [LD89], o comportamento qualitativo é semelhante em ambas as abordagens. Por outro lado, a transposição para o modelo tridimensional, apesar de não ser linear, é um passo relativamente simples, havendo apenas mais um grau de liberdade nos movimentos (o que corresponde a mais posições relativas) dos aminoácidos. Finalmente, está também provado que os resultados obtidos no modelo 2D podem ser extrapolados para o modelo 3D [LD89].

Dado que o trabalho seminal de Lau e Dill centrou-se essencialmente na apresentação do Modelo HP, das suas vantagens e desvantagens e informação que poderia ser adquirida através da sua utilização, eles usaram apenas a sequências pequenas (com um máximo de 20 aminoácidos) sobre as quais exploraram todas as conformações possíveis. Exposto o modelo e as suas potencialidades na determinação de características de sequências de aminoácidos, surgiram então (e ainda surgem) outras abordagens que evitam fazer uma busca exaustiva de todas as conformações possíveis para encontrar uma solução óptima (ou quase óptima), permitindo estudar sequências mais longas — havendo mesmo sequências com 100 aminoácidos nos testes padrão.

Resta também dizer que a opção pela utilização do modelo HP se deveu à sua relativa simplicidade (mesmo quando comparado com os modelos LPE e CGE). Ainda assim, capta a propriedade dos aminoácidos que, acredita-se, tem maior peso na conformação das proteínas; isto para além de ser o mais utilizado para validação de abordagens evolucionárias.

Seguem-se, nas próximas secções, as sequências de teste padrão mais utilizadas para testar abordagens ao modelo HP, destacando-se as mais significativas.

3.3 Sequências de Teste Padrão

Os resultados sobre o desempenho do algoritmo proposto para aplicação ao modelo HP foram encontrados aplicando-o às "sequências *Tortilla*".[6] Estas sequências de teste padrão são um conjunto de cadeias de caracteres sobre o alfabeto $\{H,P\}$. Este conjunto de sequências pode ser observado na Tab. 3.2, onde é também apresentado o tamanho das sequências e a menor energia de conformação conhecida para cada uma delas. Estas sequências são as mais utilizadas para efeitos de teste sobre o modelo HP — se bem que haja trabalhos que usam sequências próprias, principalmente antes da generalização das *"Tortilla Benchmarks"* — e foram sendo apuradas através da popularidade de alguns artigos iniciais sobre abordagens ao modelo HP. Algumas sequências de teste padrão remontam ao trabalho de R. Unger e J. Moult [UM93], e outras podem ser encontradas em [SH03] e [KBBH02]. Finalmente, há mais três sequências que podem ser encontradas em [Pel02].

As sequências na Tab. 3.2, por limitações de espaço, encontram-se em forma compacta. Assim, por exemplo, em forma extensa, a primeira sequência seria apresentada como HPHPPHHPHPPHPHHPPHPH.

Apresenta-se, de seguida, uma descrição do estado actual da arte, sendo mencionados os trabalhos que mais contribuíram para a evolução da previsão da conformação de proteínas, através da utilização de técnicas evolucionárias. Este conjunto de trabalhos não tem ambições de ser exaustivo — pesquisas *on-line* podem revelar que existe já um vasto número de trabalhos nesta área, divididos por cada uma das abordagens dentro da computação evolucionária. Procurou-se, contudo, apresentar os que mostram melhores resultados e encabeçam cada uma das respectivas abordagens evolucionárias.

[6] Este conjunto de sequências de teste padrão deve o seu nome ao *site* onde foram listadas inicialmente de uma forma agrupada, `http://www.cs.sandia.gov/tech_reports/compbio/tortilla-hp-benchmarks.html`. Apesar de esse conjunto ter sido actualizado ao longo do tempo (com a adição de novas sequências e a actualização dos melhores resultados), as suas sequências foram utilizadas em muitos dos artigos produzidos nesta área e eram muitas vezes referidas pelo epíteto "*Tortilla Benchmarks*".

Tabela 3.2: Sequências de teste padrão utilizadas no Modelo HP 2D

N.º	Sequência	Tam.	E_{min}
1	$(HP)_2PH(HP)_2(PH)_2HP(PH)_2$	20	−9
2	$H_2P_2(HP_2)_6H_2$	24	−9
3	$P_2HP_2(H_2P_4)_3H_2$	25	−8
4	$P(P_2H_2)_2P_5H_5(H_2P_2)_2P_2H(HP_2)_2$	36	−14
5	$P_2H(P_2H_2)_2P_5H_{10}P_6(H_2P_2)_2HP_2H_5$	48	−23
6	$H_2(PH_3)PH_4PH(P_3H)_2P_4(HP_3)_2HPH_4(PH)_3PH_2$	50	−21
7	$P(PH_3)_2H_5P_3H_{10}PHP_3H_{12}P_4H_6PH_2PHP$	60	−36
8	$H_{12}(PH)_2((P_2H_2)_2P_2H)_3(PH)_2H_{11}$	64	−42
9	$H_3P(PH)_3P(PH)_3P_2H$	20	−10
10	$H_4P_4H_{12}P_6(H_{12}P_3)_3HP_2(H_2P_2)_2HPH$	85	−53
11	$P(P_2H_2)_2H_2P_2H_3(PH_2)_3H_2P_8(H_6P_2)_2P_7H(PH_2)_2H_9P_2H(H_2P)_2HP(PH)_2H_2P_6H_3$	100	−48
12	$P_5(PH)_2HP_5H_3PH_5PH_2P_2(P_2H_2)_2(PH_5)_2H_5(PH_2)_2H_5P_{11}H_7P(PH)_2H_2P_5(PH)_2H$	100	−50
13	$PHP(PH)_2H_2PH_2PH_5$	18	−9
14	$(HP)_2H_3P_3H_4P_2H_2$	18	−8
15	$H_2P_5H_2P_3HP_3HP$	18	−4

3.4 Algoritmo de Monte Carlo

Uma das primeiras abordagens ao modelo HP foi o algoritmo de Monte Carlo, baseado no trabalho realizado por Metropolis *et al.* [MRRT53]. De uma forma bastante simples, este algoritmo funciona avaliando as probabilidades de distribuição de um conjunto finito de partículas sobre um espaço bem definido, usando-as, posteriormente, para propor uma determinada solução.

Observando-se o exemplo constante na Fig. 3.3, tem-se uma superfície A de área 1 e, inserida nesta, uma forma S da qual se quer determinar a área. Começa-se por distribuir aleatoriamente na superfície A um conjunto finito de pontos. De seguida, calcula-se o *ratio* entre os pontos que se encontram dentro da forma S e todos os pontos que se encontram na superfície A (S incluído). Esse *ratio* dá a proporção entre as áreas de S e A; logo, conhecendo-se a área de A, passa-se a conhecer a área de S. Obviamente, quanto maior for o conjunto de pontos gerados, mais rigorosa será a estimativa da medida.

Passando-se ao modelo HP, umas das primeiras aplicações do algoritmo de Monte Carlo foi apresentada por Unger e Moult [UM93] (onde era comparada a sua aplicação

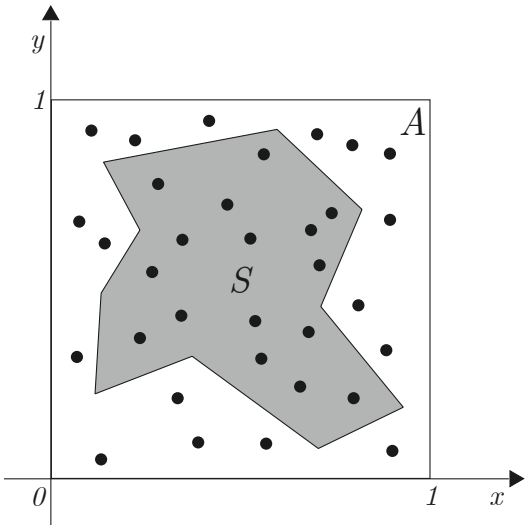


Figura 3.3: Exemplo da utilização do algoritmo de Monte Carlo para determinação da área de uma superfície

com a de algoritmos genéticos). Apesar de o exemplo antes apresentado ser bastante mais simples que a aplicação do algoritmo de Monte Carlo na previsão da conformação de proteínas, o princípio mantém-se, avaliando-se cada uma das possíveis conformações com base em probabilidades, estatística e com recurso a valores obtidos de forma aleatória. No Alg. 1 podem ser observados os passos da aplicação do algoritmo de Monte Carlo. Este algoritmo descreve-se da seguinte forma [UM93]:

1. começa-se com uma conformação aleatória C;
2. a essa conformação C, com energia E, aplica-se uma alteração aleatória única e obtém-se a conformação C', com energia E';
3. se $E' \leq E$, aceitar a nova conformação; se não, decidir de forma não-determinística se a alteração deve ser aceite ou não, de acordo com o aumento de energia resultante da alteração; o critério de aceitação é usualmente este: aceitar se $ValorAleatório < \exp\left[\frac{E-E'}{c_k}\right]$, onde $ValorAleatório$ é uma variável aleatória com uma distribuição uniforme entre 0 e 1, e onde c_k é gradualmente reduzido durante a simulação para obter convergência; se a alteração não for aceite, mantém-se a conformação inicial C;
4. se o critério de paragem não tiver sido alcançado, repetem-se os passos 2 a 4.

Nesta abordagem não são aceites conformações inválidas. De qualquer das formas, esta é uma abordagem possível e satisfatória apenas para sequências pequenas.

Procedimento Algoritmo de Monte Carlo
Entrada: conformação aleatória válida C
Saída: conformação C'
enquanto *critério de paragem não for satisfeito* **fazer**
  a partir da conformação C com energia E, criar uma única alteração aleatória nessa conformação, obtendo a conformação C' e obter a sua energia E';
  se $E' \leq E$ **então**
    aceitar a conformação C';
  senão
    se $ValorAleatório < \exp\left[\frac{E-E'}{c_k}\right]$ **então**
      aceitar a conformação C';
    senão
      manter a conformação anterior C;
    fim
  fim
fim
retornar *conformação* C'

Algoritmo 1: Algoritmo de Monte Carlo

Para além desta abordagem inicial, interessa referir outra proposta por F. Liang e W. Wong [LW01], onde voltou a ser utilizado o algoritmo de Monte Carlo com o auxilio de técnicas evolucionárias. Ao contrário da abordagem de Unger e Moult, nesta, em vez de se trabalhar com uma conformação de cada vez, fazendo-a evoluir ao longo das iterações, trabalha-se com uma população de conformações e, recorrendo a técnicas evolucionárias, são utilizados operadores de mutação e recombinação, bem como três operadores de macromutação para alteração de três padrões predefinidos pelos autores, a cada um dos elementos da população. Posteriormente, à semelhança da abordagem anterior, se a qualidade da nova população for superior à da anterior, esta é substituída; senão, tem ainda uma pequena probabilidade de ser aceite ou, com maior probabilidade, ser rejeitada. Quando proposta, esta abordagem apresentava resultados superiores ou semelhantes aos já existentes, pelo que continuou a ser um bom ponto de referência.

3.5 Algoritmo Genético

Um algoritmo genético, como foi já referido no *Capítulo 2*, é um método de pesquisa global aleatória, onde se imita o processo de evolução existente na natureza. O algoritmo genético determina, durante a sua execução, quais os indivíduos (as

soluções candidatas) que devem sobreviver, reproduzir-se ou ser afastados da população. Aplica-se o princípio da "sobrevivência dos mais aptos" para gerar soluções melhor adaptadas ao ambiente onde estão inseridas. Pode ser observada no Alg. 2 a estrutura básica de um algoritmo genético [CS04], que é partilhada em grande medida pelas implementações mais convencionais.

Função Algoritmo Genético Clássico
Entrada: problema
Saída: solução
gerar população inicial;
enquanto *não terminar* **fazer**
  avaliar população;
  seleccionar progenitores;
  gerar descendência, aplicando operadores de variação;
  substituir população actual pela descendência gerada;
fim

Algoritmo 2: Algoritmo genético clássico

Uma das primeiras abordagens com algoritmos genéticos ao modelo HP foi também realizada, tal como foi já referido, por Unger e Moult [UM93]. Nesta abordagem existe uma população de conformações válidas, criadas no início da execução do algoritmo. Depois, em cada geração, é aplicado um conjunto de mutações, à semelhança do método de Monte Carlo. Contudo, depois é aplicado um operador de recombinação, aplicado a dois progenitores seleccionados em função da qualidade relativa das suas conformações, das suas energias E_i, segundo a fórmula $p(C_i) = \frac{E_i}{\sum_{j=1}^{N} E_j}$, utilizada para ordenar os progenitores em função da sua qualidade. De seguida, os novos indivíduos são testados para verificar a sua validade, repetindo-se o processo enquanto tal não acontecer. Quando tal acontece, os indivíduos são avaliados, podendo ser aceites para incorporar a nova população, se a sua energia, E_k, for inferior à média das energias dos seus progenitores, $\bar{E}_{ij} = \frac{E_i+E_j}{2}$, $E_k \leq E_{ij}$. Se esta condição não se verificar, existe ainda a possibilidade de serem escolhidos se $ValorAleatório < \exp\left[\frac{\bar{E}_{ij}-E_k}{c_k}\right]$, nos mesmos contornos da abordagem com o algoritmo de Monte Carlo, onde $ValorAleatório$ é uma variável aleatória com uma distribuição uniforme no intervalo (0,1).

Após esta primeira abordagem com algoritmos genéticos, foram surgindo outras. A que apresentou melhores resultados até à data foi a de T. Bui e G. Sundarraj [BS05]. Essa abordagem difere da clássica de Unger e Moult em vários aspectos, nomeadamente: a representação dos indivíduos é feita com quadro direcções relativas (esquerda, direita, cima e baixo), ao invés de apenas três para codificar o posicionamento

de cada aminoácido relativamente ao anterior; é feita uma selecção de indivíduos para dois subgrupos da população que posteriormente serão utilizados para o emparelhamento dos progenitores para a recombinação; e, factor mais distinto, são utilizadas estruturas secundárias para auxiliar na obtenção das estruturas terciárias.

Sempre que é gerado um novo indivíduo, este é analisado para se verificar se pode ou não ter algumas estruturas secundárias que possam ser aplicadas — i.e., se existe na sequência alguma subsecção com algum padrão já conhecido correspondente a alguma estrutura secundária, aplicando-se essa "subconformação" à subsecção da sequência de aminoácidos. Este mecanismo é descrito no Alg. 3.

Procedimento PFGA
Entrada: instância do problema
Saída: solução candidata
gerar população inicial P;
enquanto *os critérios de paragem não forem satisfeitos* **fazer**
  seleccionar dois progenitores p_1,p_2;
  d_1,$d_2 \leftarrow$ recombinar(p_1,p_2);
  mutar descendência d_1,d_2;
  ajustar descendência d_1,d_2;
  optimizar localmente descendência d_1,d_2;
  substituir(P,p_1,p_2,d_1,d_2);
fim
retornar *o melhor membro de* P

Algoritmo 3: Algoritmo PFGA

3.6 Algoritmo Genético com Procura Tabu

Outra abordagem possível com algoritmos genéticos é a aplicação de técnicas de Procura Tabu durante a execução do algoritmo genético para a obtenção de novas soluções antes da aplicação do mecanismo de selecção e dos operadores de variação.

A procura tabu foi proposta inicialmente por Glover *et al.* ([Glo90]), sendo uma meta-heurística que pode ser aplicada a outras abordagens, para evitar que as soluções candidatas fiquem presas em mínimos locais. O método pode ser aplicado em processos que utilizem conjuntos de pequenas alterações para transformar uma solução numa outra solução, que posteriormente será avaliada.

A procura tabu descreve-se sumariamente da seguinte forma [JCSM03]: é um algoritmo meta-heurístico que mantém apenas uma solução durante o processo de

pesquisa. Primeiramente, uma solução inicial é especificada (por qualquer outro processo) ou gerada aleatoriamente no início das iterações; de seguida, são geradas algumas soluções vizinhas a partir da solução actual. Cada uma desta novas soluções é avaliada e ordenada em função dessa mesma avaliação — as melhores soluções no topo da lista e as piores no fundo —, criando-se a lista tabu. As soluções vizinhas são consideradas tabu se não forem suficientemente diferentes das constantes na lista tabu. A melhor solução vizinha será aceite se não constar da lista tabu. Contudo, se estiver na lista tabu, mas ainda assim respeitar o critério de aspiração,[7] é também aceite. A solução actual substitui uma das soluções na lista tabu. Quando uma nova solução actual é identificada e guardada, são feitas novas pesquisas de soluções vizinhas a partir dessa e é dado início a uma nova iteração. Se, após um determinado número de iterações, a melhor solução não puder ser alterada, dá-se um caso de convergência. Nesse caso, a pesquisa termina e é devolvida a melhor solução encontrada até esse momento. O papel da memória neste algoritmo — uma vez que as soluções vão sendo guardadas — é evitar que a pesquisa de soluções entre em ciclo, caindo-se num mínimo local, e continuar a pesquisar para devolver soluções próximas da óptima.

Procedimento Algoritmo Genético com Pesquisa Tabu
Entrada: instância do problema
Saída: solução candidata
definir parâmetros do algoritmo;
criar população inicial;
enquanto *a condição de paragem não for satisfeita* **fazer**
  aplicar procura tabu à população e obter a nova população válida;
  aplicar o operador de selecção e seleccionar os indivíduos (cromossomas) para a próxima geração;
  aplicar o operador de recombinação à população actual;
  aplicar o operador de mutação à população actual;
fim
retornar *a população actual*

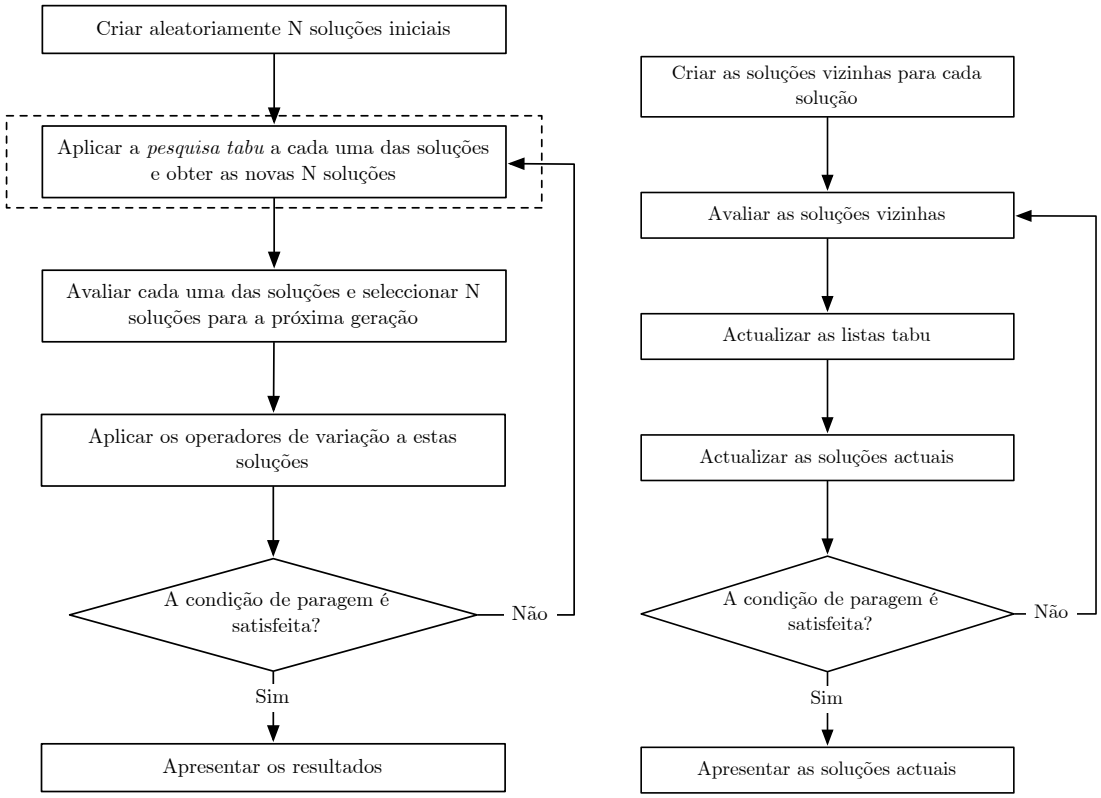
Algoritmo 4: Algoritmo genético com pesquisa tabu

A abordagem com maior visibilidade da aplicação de algoritmos genéticos com procura tabu a este problema foi feita por Jiang *et al.* [JCSM03], sendo esta descrita no Alg. 4.

[7] Critérios de aspiração são um conjunto de regras que podem estipular que uma solução, sem prejuízo de se encontrar na lista tabu, possa, mesmo assim, ser aceite. Por exemplo, se uma solução, mesmo que pertença à lista tabu, for melhor que as encontradas até ao momento, pode incorporada na população.

O método de procura tabu, nesta abordagem, é assim descrita: as soluções vizinhas de uma solução são obtidas através da mutação aleatória de k genes, onde k é um valor aleatório entre 1 e o tamanho da sequência. Os k genes são também escolhidos aleatoriamente. Depois, o operador de procura tabu é aplicado a cada uma das soluções actuais. Após uma iteração, as melhores soluções actuais são apresentadas.

Pode ser observada na Fig. 3.4 uma esquematização do algoritmo genético e do passo em específico onde é aplicada a procura tabu.



(a) Fluxograma de um algoritmo genético híbrido (b) Fluxograma de uma procura tabu

Figura 3.4: Fluxograma de um algoritmo genético com procura tabu

3.7 Algoritmo Memético

Outra abordagem, que pode ser considerada também uma variante dos algoritmos genéticos (especialmente daqueles com pesquisa local), é a utilização de algoritmos meméticos. Enquanto que os algoritmos genéticos foram inspirados na evolução biológica, os algoritmos meméticos procuram imitar a evolução cultural [Mos89].

De acordo com [KS05], num algoritmo memético cada elemento da população realiza uma pesquisa local, após a qual interage com os outros indivíduos, trocando informação. É desta partilha informação (não genética), que surge o termo *meme*, em analogia com o termo gene, onde um contém informação cultural e o outro contém informação genética [Daw06]. Desta interacção entre indivíduos vai resultar a criação de um novo indivíduo, regra geral através do mecanismo de recombinação. A pesquisa local e troca de informação entre indivíduos vai acontecendo em cada iteração até que o critério de paragem seja atingido.

Nos Algs. 5 e 6, retirados de [KS05], podem ser observados o método de pesquisa local e o esquema de um algoritmo memético, respectivamente. A utilização do mecanismo de pesquisa local é partilhado por várias abordagens distintas; no entanto, a partilha de *memes* só acontece nos algoritmos meméticos, sendo este o elemento diferenciador.

Procedimento Pesquisa Local Padrão
Entrada: instância do problema
Saída: solução candidata
criar uma solução inicial s para instância x do problema;
enquanto *não for encontrado um óptimo local* **fazer**
  gerar o próximo vizinho $n_{x,s}$, utilizando s e x;
  se $n_{s,x}$ *for melhor que* s **então**
    $s = n_{s,x}$;
  fim
fim
retornar s

Algoritmo 5: Pesquisa local padrão

Procedimento Algoritmo Memético
Entrada: instância do problema
Saída: solução candidata
iniciar aleatoriamente a população de *Progenitores*;
enquanto *a condição de paragem não for satisfeita* **fazer**
  aplicar *pesquisa local* aos *Progenitores*;
  seleccionar *Progenitores* para uma *"pool" de acasalamento*;
  obter a descendência através da recombinação da *"pool" de acasalamento*;
  criar os novos *Progenitores* através de uma *selecção no conjunto dos Progenitores e descendência*;
fim
retornar *a população actual*

Algoritmo 6: Algoritmo memético

Apesar de não ter sido encontrada uma abordagem com algoritmos meméticos aplicada ao modelo HP com resultados considerados relevantes, há uma abordagem com uma variante, com *multimemes*, que apresenta resultados interessantes. Segundo os autores (Krasnogor *et al.* [KBBH02]), um algoritmo multimeme diferencia-se de um algoritmo memético na medida em que, em vez de possuir apenas um método heurístico de pesquisa local, possui um conjunto deles, do qual é seleccionado um de forma autoadaptativa durante a execução do algoritmo, de acordo com a instância do problema utilizada, com o estádio da procura, ou com os indivíduos na população.

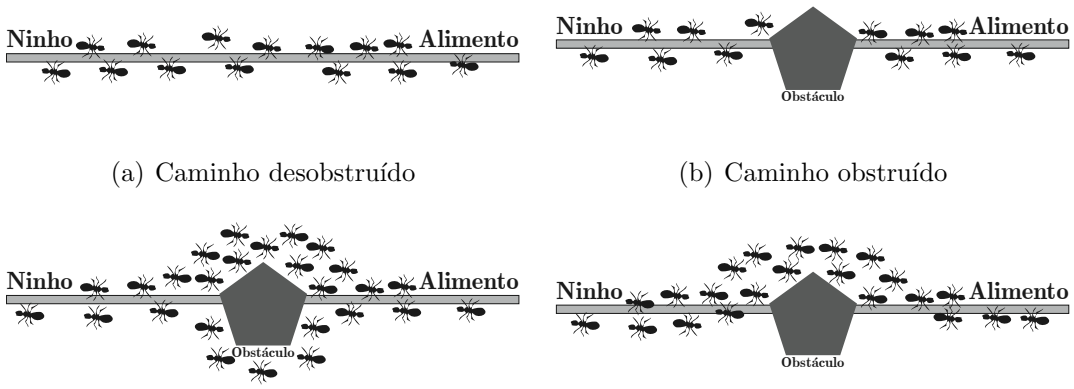
Parafraseando ainda Krasnogor *et al.* [KBBH02], segue-se a descrição de um algoritmo multimeme. Neste algoritmo, um indivíduo é composto por material genético e material memético. Quanto ao material genético, este é transmitido usando os tradicionais mecanismos de recombinação e mutação. Relativamente ao material memético, a transmissão é feita através de memes (operadores de busca local) transportados por um dos progenitores. Se o meme for comum aos progenitores, este é passado ao descendente; senão, é passado o meme do progenitor melhor adaptado ou, se os progenitores tiverem a mesma avaliação, é feita uma selecção aleatória de um dos deles. A intenção é favorecer os memes associados aos indivíduos mais aptos. Também, para gerar diversidade, o meme de um indivíduo pode ser substituído por um outro seleccionado de um conjunto predefinido de memes, durante o processo de mutação.

3.8 Optimização por Colónia de Formigas

A Optimização por Colónia de Formigas[8] baseia-se no comportamento de colónias de formigas reais, tendo sido inicialmente proposto por Dorigo *et al.* [DMC96]. A ideia por detrás deste algoritmo é a de que as formigas (agentes, neste caso) trocam informação entre si sob a forma de rastos de feromona. Cada formiga cria um trilho desde o seu ninho até à fonte de alimento ou, neste caso, cada agente explora cada uma das soluções, atribuindo-lhe um determinado peso (o rasto de feromona). Esse trilho é também avaliado em função da rapidez com que é percorrido e do número de vezes que é percorrido. Assim, sempre que um agente volta a explorar essa solução, reforça o seu rasto, o que levará a que outros agentes façam o mesmo até todos os agentes percorrerem o mesmo caminho.

Observe-se a Fig. 3.5 onde é descrita a situação em que um obstáculo surge no meio de um caminho de formigas entre o ninho e a fonte de alimento. Inicialmente

[8]Em inglês, *Ant Colony Optimization* (ACO).



(a) Caminho desobstruído

(b) Caminho obstruído

(c) Dois novos caminhos

(d) Um único caminho

Figura 3.5: Caminhos de formigas

(Fig. 3.5(a)), as formigas percorrem todas o mesmo caminho. Num segundo momento (Fig. 3.5(b)) surge um obstáculo no meio do caminho e as formigas terão de decidir aleatoriamente por onde seguir, por um ou por outro lado do obstáculo, sendo que probabilisticamente, à partida, metade das formigas tomará um dos caminhos e a outra metade o outro. Contudo, como um dos caminhos é mais curto que o outro, mais formigas passarão por esse caminho no mesmo espaço de tempo que levam a percorrer o outro caminho, deixando um rasto de feromona mais intenso (Fig. 3.5(c)), levando a que as formigas, quando têm decidir novamente qual dos caminhos devem percorrer, escolham aquele com o rasto mais intenso — o que se traduz em ser o caminho mais curto. Finalmente (Fig. 3.5(d)), todas as formigas acabam por adoptar o caminho mais curto. Existe também um factor de evaporação do rasto de feromona, o que leva a que o caminho menos utilizada seja eventualmente completamente esquecido.

Procedimento Optimização por Colónia de Formigas para optimização estática
Entrada: instância do problema
Saída: solução candidata
inicializar os rastos de feromonas;
enquanto *os critérios de paragem não forem satisfeitos* **fazer**
    construir as soluções candidatas;
    realizar pesquisa local;
    actualizar os rastos de feromonas;
fim
retornar *a melhor solução encontrada*

Algoritmo 7: Algoritmo genérico de uma optimização por colónia de formigas

A abordagem com maior visibilidade sobre a aplicação do método de optimização por colónia de formigas ao problema em estudo foi a proposta por A. Shmygelska e H. Hoos [SH03]. Esta abordagem segue o esquema genérico de um algoritmo para optimização por colónia de formigas, que pode ser observado no Alg. 7, mas acrescenta-lhe dois métodos probabilísticos de pesquisa local iterativa, um com melhoramento iterativo e outro com melhoramento iterativo probabilístico, que podem ser observado nos Alg.s 8 e 9, respectivamente.

Procedimento Pesquisa Local com melhoramento iterativo
Entrada: conformação c
Saída: conformação c'
enquanto *os critérios de paragem não forem satisfeitos* **fazer**
  escolher aleatoriamente o índice i uniformemente de um valor entre 1 e n;
  $c' =$ movimentoDeAlcanceLongo(i);
  se $E(c') \leq E(c)$ **então**
    retornar c'
  senão
    retornar c
  fim
fim

Algoritmo 8: Procedimento de pesquisa local com melhoramento iterativo

Procedimento Pesquisa Local com melhoramento probabilístico iterativo
Entrada: conformação c
Saída: conformação c'
enquanto *os critérios de paragem não forem satisfeitos* **fazer**
  escolher aleatoriamente o índice i uniformemente de um valor entre 1 e n;
  $c' =$ movimentoDeAlcanceLongo(i);
  se $E(c') \leq E(c)$ **então**
    retornar c'
  senão
    se Aleatório(0,1) $\leq \frac{E(c')}{E(c)}$ **então**
      retornar c'
    senão
      retornar c
    fim
  fim
fim

Algoritmo 9: Procedimento de pesquisa local com melhoramento probabilístico iterativo

3.9 Resultados conhecidos das abordagens apresentadas

Na Tab. 3.3, adaptada de [BS05], pode observar-se a comparação entre os melhores resultados obtidos pelas abordagens descritas aplicadas às sequências de teste padrão (as *Tortilla Benchmarks*),[9] nomeadamente com as aqui retratadas: a PFGA *(Protein Folding Genetic Algorithm)* [BS05], a ACO *(Ant Colony Optimization)* [SH03], a EMC *(Evolutionary Monte Carlo)* [LW01], a GTS *(Genetic algorithm combined with Tabu Search)* [JCSM03], a MMA *(Multimeme Algorithm)* [KBBH02], a MMC *(Metropolis Monte Carlo)* [UM93] e a GA *(Genetic Algorithm)* [UM93].

Tabela 3.3: Melhores resultados obtidos pelas diferentes abordagens constantes no estado da arte às sequências de teste padrão estudadas

Seq.	E_{min}	**PFGA**	**ACO**	**EMC**	**GTS**	**MMA**	**MMC**	**GA**
1	**−9**	**−9**	**−9**	**−9**	**−9**	**−9**	**−9**	**−9**
2	**−9**	**−9**	**−9**	**−9**	**−9**	–	**−9**	**−9**
3	**−8**	**−8**	**−8**	**−8**	**−8**	**−8**	**−8**	**−8**
4	**−14**	**−14**	**−14**	**−14**	**−14**	**−14**	−13	−12
5	**−23**	**−23**	**−23**	**−23**	**−23**	−22	−20	−22
6	**−21**	**−21**	**−21**	**−21**	**−21**	**−21**	**−21**	**−21**
7	**−36**	**−36**	**−36**	−35	−35	**−36**	−33	−34
8	**−42**	**−42**	**−42**	−39	−39	−38	−35	−37
9	**−10**	–	–	–	–	–	–	–
10	**−53**	**−53**	−51	–	–	–	–	–
11	**−48**	**−48**	−47	–	–	–	–	–
12	**−50**	−49	−47	–	–	–	–	–
13	**−9**	–	–	–	–	–	–	–
14	**−8**	–	–	–	–	–	–	–
15	**−4**	–	–	–	–	–	–	–

Pode ser observado, na Tab. 3.3, que a abordagem que apresenta melhores resultados nas sequências de teste padrão é a PFGA. As abordagens GA e MMC (as aplicações mais simples com utilização de um algoritmo genético e de um algoritmo de Monte Carlo) apresentam os resultados mais baixos — o que não é de estranhar, dadas as características específicas do problema. À medida que as abordagens começam a

[9] Nas sequências de teste padrão para as quais não são conhecidos os melhores resultados de determinada abordagem, foi colocado um traço ("–").

ser mais complexas, como é o caso da EMC (que aplica técnicas evolucionárias ao algoritmo de Monte Carlo), da GTS (que aplica a pesquisa tabu ao algoritmo genético) e da MMA (com um algoritmo multimeme), os resultados passam a ser melhores. Pode-se constatar assim que, de facto, o problema não é trivial, levando à adopção de abordagens mais complexas. Finalmente, as melhores abordagens são a PFGA e a ACO, onde, entre outras técnicas, se faz uso de técnicas de pesquisa e optimização local para auxiliar a evolução da população.

3.10 Outras Abordagens

Há ainda abordagens que interessa mencionar porque, por algum motivo, influenciaram parcialmente o trabalho desenvolvido. Algumas não foram aplicadas às sequências de teste padrão, não fornecendo assim pontos de comparação; outras têm algumas especificidades que levam a que não sejam consideradas, também, como bons pontos de comparação. Passam a ser descritas de seguida essas abordagens.

3.10.1 PERM

Uma das abordagens mencionadas em vários artigos pelos seus resultados (por exemplo, em [BS05]), é a "*Pruned-Enriched Rosenbluth Method*",[10] proposta por Bastolla *et al.* [BFG+98]. Esta abordagem é baseada no algoritmo de Monte Carlo e no método Rosenbluth-Rosenbluth e descreve-se, sucintamente, desta forma: os aminoácidos vão sendo colocados sequencialmente em posições vagas, com uma distribuição probabilística; a cada conformação é atribuído um determinado peso P; à medida que os pesos vão aumentando a par com o tamanho das cadeias, é dado início ao processo de poda, removendo-se as conformações com pouco peso e substituindo-as por conformações com um peso maior. A poda é feita de forma estocástica: se o peso de uma determinada conformação desce abaixo de um determinado patamar $P^{<}$, é eliminada com uma probabilidade de $\frac{1}{2}$, ao passo que é mantida e o seu peso duplicado na outra metade dos casos. O "enriquecimento" é feito de forma independente: se P subir acima de um outro patamar $P^{>}$, a configuração é substituída por n cópias, cada com um peso $\frac{P}{n}$.

[10] Este método pode ser traduzido livremente, para português, como "Método Rosenbluth Enriquecido por Poda".

Contudo, esta abordagem, apesar de mostrar resultados bastante bons para algumas das sequências de teste padrão — das sequências utilizadas, podem ser comparadas a 7, a 11 e a 12, tendo obtido resultados de -35, -47 e -49, respectivamente —, nunca teve grande expressão. Tal é devido à sua inferior adaptação de uma sequência para outra sequência — i.e., à medida que se vão escolhendo sequências de teste padrão para análise, o programa tem um comportamento díspar e, assume-se, pouco escalável. Assim, é por vezes necessária a adopção de pequenos truques — *"special tricks"*, utilizando a expressão dos próprios autores [BFG+98] — para adaptação do algoritmo, conforme as sequências em estudo, de forma a obter os resultados esperados. Ora, na melhor das hipóteses, espera-se que a parametrização possa ser ajustada, nunca a estrutura interna do programa (ou algoritmo), conforme a sequência em estudo.

3.10.2 Macromutações

Sobre a utilização de macromutadores, há a dizer que estes não são um mecanismo apenas aplicável a este problema em específico, nem é este mecanismo que distinguirá uma determinada abordagem — será apenas um dos aspectos envolvidos (até algumas das abordagens já mencionadas têm alguma espécie de macromutador ou operador de mutação específico). Contudo, há a convicção, expressa, por exemplo, por Krasnogor *et al.* em [KHSP99], de que os operadores de macromutação podem agir como um poderoso método de pesquisa local — especialmente aqueles que funcionam com busca e substituição de padrões.

Conforme a abordagem, por vezes surgem diferentes propostas de operadores de macromutação. No entanto, alguns dos mais comuns são a "manivela", a "dobragem", a "serpenteação" ou a "vincagem", que podem ser encontrados descritos num *poster* por Rylance *et al.*, em [RCMJJ04].

3.10.3 Indivíduos Inválidos e Penalizações

Apesar de nenhuma das abordagens já apresentadas aceitarem indivíduos inválidos, continuando, regra geral, a gerar indivíduos até que todos sejam válidos, não é inédita uma abordagem onde sejam aceites indivíduos inválidos.

A ideia por detrás da aceitação de indivíduos é o princípio de que o próprio algoritmo tratará de favorecer os melhores indivíduos (válidos) em detrimento dos piores

indivíduos (inválidos), mantendo a diversidade dos indivíduos e diminuindo o tempo de execução do algoritmo (já que não é necessário aguardar que todos os indivíduos válidos sejam gerados, quer na população inicial, quer na geração da descendência).

Contudo, é necessário contrabalançar a aceitação de indivíduos inválidos com um mecanismo que os permita diferenciar dos indivíduos válidos. Tal é conseguido com a adopção de penalidades a conformações que tenham aminoácidos sobrepostos — i.e., a ocupar a mesma posição — sendo assim, portanto, conformações inválidas. Em geral, estas penalidades são proporcionais ao número de aminoácidos sobrepostos. Desta forma, o algoritmo é "forçado" a afastar as conformações piores, favorecendo as melhores e, ao mesmo tempo, válidas. Exemplos deste mecanismo podem ser encontrados em [HYTY05] e [PPG95] (abordagens ao modelo HP tridimensional) ou em [Rat04] (no modelo bidimensional).

A abordagem apresentada em [Rat04], por V. Ratakonda, apenas apresenta resultados para as sequências de teste padrão 1, 3, 4, 5, 6 e 8, tendo igualado os melhores resultados, à excepção da sequência 5, onde fica a um ponto do melhor resultado conhecido (-23). Para além da adopção de indivíduos inválidos, esta abordagem também faz uso do sistema de fundo de progenitores e da execução em paralelo com várias populações em simultâneo.

Capítulo 4

Método Proposto

O algoritmo genético utilizado para a resolução do problema estudado nesta dissertação segue algumas das linhas comuns no algoritmo genético clássico, possuindo como características diferenciadoras o facto de utilizar um mecanismo de reparação de indivíduos inválidos, taxas de variação dinâmicas, bem como a utilização de técnicas de mutação específicas ao problema.

Por diversos motivos, entre os quais a sua identificação para efeitos de comparação com outras abordagens, foi necessário dar um nome a esta abordagem e, por várias razões de ordem prática, optou-se por um nome em inglês para a abordagem proposta: GARMM *(Genetic Algorithm with a Repair Mechanism and Macromutations).*

Nas secções seguintes deste capítulo são abordados o algoritmo genético proposto, bem como as suas especificidades na aplicação ao Modelo HP.

4.1 Abordagem ao Modelo HP

Nesta dissertação optou-se por um modelo bidimensional (o Modelo HP 2D) para testar a abordagem evolucionária ao problema da previsão da conformação de proteínas. Apesar de já existirem abordagens evolucionárias a este problema, procurou-se uma variante na qual, ao invés de se descartar os indivíduos inválidos gerados (que, à medida que as sequências de teste aumentam o seu tamanho, passam a constituir a grande maioria descendência gerada) ou de os penalizar severamente (levando a que a sua expressão seja mínima), se procura reparar os indivíduos inválidos.

Apesar de na maior parte das abordagens os indivíduos inválidos serem descartados, como acaba por ser o caso da maioria das expostas no *Capítulo 3*, existem já outras onde se toma em conta os indivíduos inválidos — veja-se, por exemplo, [PPG95].

No entanto, nestas abordagens, os indivíduos são apenas mantidos com o intuito de manter diversidade na população. Tanto quanto se sabe, neste tipo de aplicações nunca se aplicou aos indivíduos inválidos um mecanismo de reparação.

À semelhança das abordagens mais comuns, os indivíduos foram definidos com direcções relativas definidas no alfabeto $\{L,F,R\}$, assumindo-se também que a primeira direcção se encontra predefinida. Deste modo, o tamanho do genótipo é menor em duas unidades que a sequência HP.

4.2 Algoritmo Genético

O algoritmo genético utilizado como base do programa segue, em grande medida, o método tradicional dos algoritmos genéticos clássicos (ver Alg. 2). No entanto, dado o problema em estudo, várias alterações foram necessárias devido a factores como a representação dos indivíduos, a função de avaliação e as condições de paragem.

As várias alterações ao algoritmo genético clássico, propostas neste trabalho, prendem-se com:

1. não substituição integral da população pela descendência gerada, havendo competição entre os novos indivíduos (a descendência) e os seus progenitores pelos lugares na nova população;

2. utilização de operadores de macromutação (mutações específicas ao problema em estudo, aplicadas a conjuntos de genes);

3. alteração dinâmica das taxas de variação a aplicar aos indivíduos, ora aumentando-se a taxa de recombinação e diminuindo as taxas de mutação e macromutação, ora diminuindo a taxa de recombinação e aumentando as taxas de mutação e macromutação (de forma a evitar convergência prematura da população e a gerar diversidade em momentos específicos da execução do algoritmo);

4. utilização de elite (salvaguardando-se os melhores indivíduos de cada geração); neste programa, quando há vários indivíduos, distintos, com a mesma pontuação, estes são escolhidos de forma aleatória, de acordo com a ordenação dos indivíduos nesse instante;

5. utilização de um mecanismo de reparação de indivíduos (ver Alg. 10).

São também, no final da execução do algoritmo, devolvidos os melhores indivíduos, e não apenas o melhor indivíduo, visto poder haver mais do que uma solução com a mesma energia mínima de conformação para um mesmo problema.[1]

O mecanismo de reparação revela ter especial interesse e aplicabilidade devido ao facto de, no problema estudado, muitos dos indivíduos gerados se revelarem inválidos — i.e., possuem uma conformação impossível, havendo dois ou mais aminoácidos a ocupar uma mesma posição. No entanto, por vezes dá-se o caso de determinado indivíduo representar uma solução com elevado potencial, sendo inválido devido a alguma sobreposição de aminoácidos em pontos "não críticos" ou "fáceis de corrigir". Seria assim uma boa abordagem tentar reparar esse mesmo indivíduo, ao invés de pura e simplesmente descartá-lo, havendo a possibilidade de se obter uma melhor solução.

Segue-se, nas próximas subsecções, uma descrição dos aspectos mais relevantes do algoritmo genético, onde são focados a representação dos indivíduos, a constituição da população, o funcionamento da função de avaliação, o método de selecção adoptado, a implementação dos operadores de variação, o mecanismo de reparação, a especificação da condição de paragem e, finalmente, a enunciação dos parâmetros utilizados na configuração do algoritmo genético. É dada especial ênfase ao mecanismo de reparação.

4.2.1 Representação dos Indivíduos

A representação dos indivíduos consiste no conjunto de direcções que são utilizadas para definir as posições relativas dos aminoácidos numa proteína, em função dos anteriores: L (*left* — esquerda), F (*forward* — frente) e R (*right* — direita). Na prática, são usadas cadeias de números inteiros com três valores distintos: -1, 0 e 1. Assim, optou-se por uma representação de um número inteiro por direcção, por esta poder apenas assumir um de três valores. Com uma abordagem com apenas três direcções relativas, há uma situação passível de gerar indivíduos inválidos que é evitada logo à partida: não é possível recuar para a posição onde se encontra o aminoácido anterior.

Como foi já referido, o tamanho das cadeias é igual ao número de aminoácidos da proteína menos dois. Com efeito, entre quaisquer dois aminoácidos existe apenas

[1] Na prática, a existirem, só são apresentadas várias soluções quando é utilizada o interface gráfico para o utilizador (GUI). Quando executado sem o GUI, por questões de ordem prática, o programa apenas guarda a solução no topo das melhores soluções, mesmo que haja várias com a mesma pontuação.

Entrada: sequência HP
Saída: melhor(es) indivíduo(s) da população
gerar população inicial;
enquanto *número máximo de gerações não for atingido* **e** *número máximo de gerações sem evolução dos melhores indivíduos não for atingido* **fazer**
  avaliar população;
  se *patamar de estagnação alcançado* **então**
    actualizar valores das **taxas de variação**;
  senão
    repor valores base das taxas de variação;
  fim
  seleccionar melhores indivíduos para passarem inalterados à geração seguinte, criando uma **elite**;
  seleccionar possíveis progenitores para reprodução de acordo com o seu mérito, criando *pools*;
  gerar descendência: aplicar o operador de **recombinação** aos progenitores;
  aplicar os operadores de **macromutação** à descendência;
  se *não foi executada uma macromutação* **então**
    aplicar o operador de **mutação**;
  fim
  se *descendência inválida* ***e*** *acima do patamar de reparação* **então**
    reparar descendência;
  fim
  juntar população antiga e descendência, seleccionando os melhores indivíduos e adicionando-lhes depois os elementos da elite;
fim

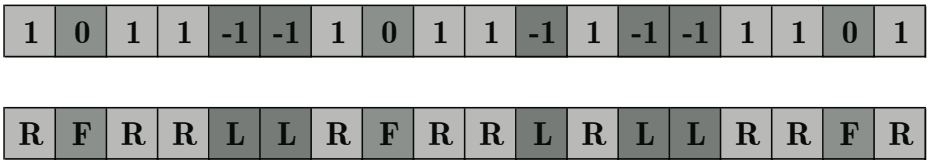
Algoritmo 10: Algoritmo genético utilizado

uma direcção, sendo também que por definição a primeira direcção numa de cadeia de aminoácidos é sempre constante (para evitar resultados idênticos em que os indivíduos apenas diferem uns dos outros por rotação relativamente à origem).

Na Fig. 4.1(a) encontra-se a representação linear da sequência HP, já apresentada na Fig. 3.1, e na Fig. 4.1(b) encontra-se a representação do genótipo de um indivíduo utilizando as mnemónicas das direcções (L, F e R) e os valores inteiros utilizados na representação interna ao programa. O genótipo (Fig. 4.1(b)), aplicado à sequência HP (Fig. 4.1(a)), permite a obtenção do fenótipo apresentado na Fig. 3.1. Mais uma vez, note-se a diferença de tamanho, em duas unidades, existente entre a quantidade de aminoácidos e a de direcções: o número de aminoácidos é superior em um ao número de direcções; e fixa-se a primeira direcção (tendo sido arbitrada a direcção F), não sendo assim necessária a sua representação no genótipo.

Os indivíduos são gerados de forma aleatória no início da execução do programa,

(a) Sequência de aminoácidos hidrofóbicos e polares



(b) Sequência de genes no genótipo do indivíduo

Figura 4.1: Representação dos aminoácidos e dos indivíduos

sendo que nesta fase, apesar de serem admitidos indivíduos válidos, todos os indivíduos devem possuir uma energia mínima de conformação inferior ou igual a zero. À partida, todos os indivíduos válidos respeitam esta condição, já que não possuem sobreposições e, logo, nunca podem ter energias de conformação positivas — quando muito, serão nulas; já os indivíduos inválidos, devido às penalidades que podem sofrer, podem ou não respeitar esta condição, sendo, no último caso, reparados.

O propósito desta condição é não enviesar logo à partida o programa com soluções que podem ser consideradas bastante más. É de notar que a partir de quatro aminoácidos a probabilidade de serem gerados indivíduos inválidos passa a ser maior que zero e vai aumentando com o tamanho da sequência HP em estudo, ultrapassando facilmente a probalidade de geração de indivíduos válidos. A título de exemplo, considere-se o caso mais simples para que uma sobreposição ocorra. Para tal, basta que haja quatro direcções que sejam todas *esquerda* (ou todas *direita*) para que o quinto aminoácido se posicione sobre o primeiro, dando origem a uma conformação inválida.

Depois de gerada a população inicial, já passam a ser aceites indivíduos inválidos que não respeitem essa condição, competindo directamente com os indivíduos válidos da população. No entanto, abaixo de um determinado patamar, calculado em função da pontuação do melhor indivíduo gerado até ao momento, os indivíduos passam a ser sujeitos a reparação. Este patamar, apurado de forma empírica, corresponde a 62,5% da pontuação do melhor indivíduo apurado até essa iteração: os indivíduos cuja energia mínima de conformação seja inferior ou igual a esse patamar mantêm-se inalterados na população, apesar de inválidos, assumindo-se que indivíduos abaixo desse patamar possuem apenas ligeiras incorrecções que poderão "desaparecer" ou mesmo contribuir para a constituição de um melhor indivíduo; os indivíduos cuja

energia mínima de conformação seja superior a esse patamar são eleitos para reparação, assumindo-se que as incorrecções poderão influenciar negativamente futuros indivíduos, ao enviesar futuros desenvolvimentos da população, sendo a melhor alternativa a sua reparação.

Outra restrição colocada aos indivíduos inválidos é estes não poderem ser considerados como candidatos ao melhor indivíduo da população — a solução proposta nunca pode ser inválida, sob pena de não ter qualquer utilidade e desperdiçar preciosos recursos.

4.2.2 População

A população é o conjunto de todos os indivíduos num determinado instante de tempo da execução do algoritmo, sendo ela que está sujeita ao processo de evolução. A evolução só se pode verificar se existir competição entre os indivíduos e interacção destes com o meio-ambiente. Parafraseando Darwin, a evolução faz-se pela sobrevivência dos mais aptos. Assim sendo, é sempre necessária uma população com indivíduos que possam competir entre si, utilizando algum método de selecção, para que haja evolução, seja através de mutação, de recombinação ou de ambos.

Numa população, uma das possíveis medidas de qualidade é a sua diversidade, principalmente no estágio inicial de execução do algoritmo, onde se pretende uma procura ampla do espaço de soluções. Enquanto houver indivíduos distintos dentro de uma população, a probabilidade de gerar novos indivíduos (também eles distintos) por recombinação de progenitores é bastante elevada. À medida que a diversidade vai diminuindo — o que, regra geral, é indicação de que houve convergência e de que um dos indivíduos está a "tomar" toda a população — a probabilidade de gerar novos indivíduos distintos por recombinação reduz-se drasticamente, anulando-se mesmo quando todos os indivíduos são iguais. Nessa situação, a geração de *novos* indivíduos passa a estar dependente, sobretudo, dos operadores de mutação. Devido a isto, uma das alterações ao algoritmo genético clássico é a utilização de taxas dinâmicas de variação, tornando a evolução ora mais dependente dos mecanismos de recombinação, ora mais dependente dos mecanismos de mutação.

Ao longo das várias gerações, a população actual e a sua descendência são unidas, passando os seus elementos a competir por um lugar na nova população a formar. Finalmente, depois de formada a nova população (baseada na população anterior e respectiva descendência), são adicionados os elementos que se encontram na elite.

4.2.3 Função de Avaliação

A avaliação de um indivíduo é feita com base na qualidade do seu fenótipo. O fenótipo dos indivíduos conhece-se quando se aplicam as direcções (representadas no seu genótipo) aos aminoácidos, em conjunto com a sequência de propriedades hidrofóbicas e polares que definem os aminoácidos da proteína em estudo, obtendo-se a conformação desse indivíduo. Posto isto, cada par não adjacente de aminoácidos hidrofóbicos, H, que se encontrem juntos, diminui em uma unidade a energia mínima de conformação. No início de cada avaliação, o valor base de energia para indivíduo é zero. Por outro lado, no caso de indivíduos inválidos serem aceites na população, sempre que dois aminoácidos se encontram sobrepostos — i.e., as direcções codificadas no genótipo do indivíduo levam a que dois aminoácidos fiquem a ocupar uma mesma posição —, o indivíduo é penalizado com a atribuição de pontos de energia positiva, levando a que a energia de conformação aumente.

Pontuação e Penalização

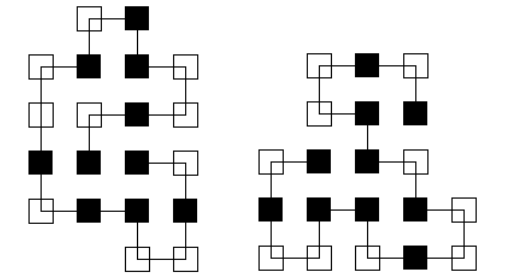
Os sistemas de pontuação e penalização são um aspecto fulcral da função de avaliação. Por omissão, e como proposto originalmente [LD89], por cada contacto H–H é atribuído -1 ponto de energia.

Quanto ao sistema de penalização, há vários valores propostos dentro dos vários trabalhos que aceitam indivíduos inválidos; no entanto, todos eles são valores positivos baixos, com o intuito de penalizar os indivíduos inválidos, mas não de os penalizar excessivamente, levando a que, em termos práticos, estes sejam excluídos da população [PPG95]. Por outro lado, optando-se pela alternativa de aceitar indivíduos inválidos na população, é necessário que estes sofram uma qualquer penalidade pela sobreposição de aminoácidos, para evitar o risco (elevado) de indivíduos inválidos tomarem controlo de toda a população. Assim, os sistemas de penalização adoptam valores superiores ou iguais a 1 ponto por cada sobreposição. Existem implementações que utilizam um sistema de penalidade variável [KHSP99]; no entanto, o mais usual é utilizar um valor fixo. Neste trabalho, o valor que, por análise empírica, se revelou mais ajustado foi 2, sendo este o mais comum a diversas abordagens, como em [PPG95] ou [Rat04].

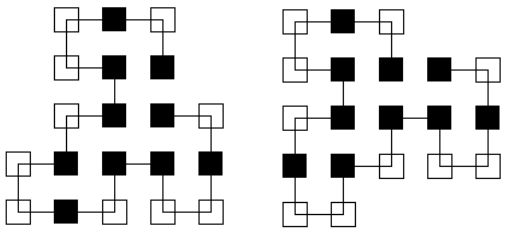
A título de exemplo, na Fig. 4.2 podem ser observados alguns dos melhores indivíduos (já que há vários com a mesma pontuação) para penalidades de 0, 1, 2 e 3

pontos, para a sequência HPHPPHHPHPPHPHHPPHPH, cuja melhor solução apurada tem uma energia mínima de conformação de -9 unidades, com uma população de 500 indivíduos e para um mínimo de 1000 gerações, para um mínimo de 3 execuções do programa. Note-se que:

- na Fig. 4.2(a) a energia mínima de conformação é de -8 pontos, tendo-se obtido como melhor valor -13, mas num indivíduo inválido;[2]
- na Fig. 4.2(b) a energia de conformação é de -6 pontos, tendo-se obtido como melhor valor -10, mas num indivíduo inválido;
- na Fig. 4.2(c) a energia de conformação é de -9 pontos, o melhor resultado, tendo-se obtido como melhor valor -9, já num indivíduo válido;
- na Fig. 4.2(d) a energia de conformação é de -9 pontos, tendo-se obtido como melhor valor -9, também num indivíduo válido.



(a) Penalidade = 0 (b) Penalidade = 1



(c) Penalidade = 2 (d) Penalidade = 3

Figura 4.2: Soluções propostas para a sequência HPHPPHHPHPPHPHHPPHPH com diversas penalidades

Apesar de o exemplo apresentado ser apenas um caso de complexidade menor, pode-se observar que aceitar indivíduos inválidos sem aplicar qualquer penalidade,

[2]Os valores de energia obtidos para indivíduos inválidos, neste caso e nos seguintes, apenas são observáveis nos dados de saída do programa, em modo verboso, não disponível no interface do programa utilizado por omissão.

Fig. 4.2(a), ou com uma penalidade de apenas uma unidade, Fig. 4.2(b), leva a que os indivíduos inválidos tomem controlo da população, como se pode verificar pelas energias mínimas de conformação, -13 e -10, respectivamente, e que afecta o desenvolvimento de indivíduos válidos, que se ficaram por energias mínimas de conformação de -8 unidades. Já com penalidades de dois e três pontos, Figs. 4.2(c) e 4.2(d), respectivamente, os indivíduos inválidos têm um peso semelhante aos dos indivíduos válidos. A partir de três pontos de penalização, os indivíduos inválidos começam a ser severamente penalizados e deixam de ter um peso relevante na constituição da população. Este efeito é ainda mais visível em cadeias mais extensas. Com tudo isto, se com uma penalidade de dois pontos se consegue o resultado pretendido, estipula-se que não existe qualquer vantagem em usar uma penalidade mais elevada.

4.2.4 Método de Selecção

Sendo este um algoritmo geracional onde se aplica o mecanismo recombinação, houve situações onde foi necessário utilizar um método de selecção. O método escolhido foi a selecção por ordem,[3] onde a probabilidade de cada indivíduo ser seleccionado é obtida com recurso a uma distribuição geométrica. Uma distribuição geométrica é uma distribuição probabilística do número de tentativas falhadas $Y = X - 1$ antes do primeiro evento com sucesso, cujo domínio é o conjunto $\{0,1,2,3,\ldots\}$. Se a probabilidade de sucesso de cada tentativa é p, a probabilidade de que sejam necessárias k tentativas é dada pela expressão $P(X = k) = (1-p)^{k-1}p$, para $k = 1,2,3,\ldots$. No entanto, na prática, como existe um cúmulo de valores para se obter o indivíduo seleccionado, utiliza-se função de distribuição cumulativa $P(Y = k) = 1 - (1-p)^k$, $k \in \{0,1,2,3,\ldots\}$.

Optou-se pela selecção por ordem, com utilização de uma distribuição geométrica, porque esta acaba por dar a todos os elementos uma probabilidade, por mais pequena que seja, de virem a fazer parte dos indivíduos seleccionados. Dá, no entanto, probabilidades maiores aos indivíduos no topo da lista, mas não tanto que estes sejam seleccionados sistematicamente, de forma a fazer evoluir a própria população com base nos seus melhores representantes garantindo-se, ao mesmo tempo, a existência de diversidade.

Assim, na selecção por ordem os indivíduos são ordenados de acordo com o seu mérito: aqueles com menor energia de conformação no início da lista e aqueles com

[3] *Ranking selection*, em inglês.

maior energia de conformação no fim. Indivíduos com o mesmo valor de energia são ordenados de acordo com a ordem em que vão sendo encontrados, o que pode variar de geração para geração. Depois, de acordo com a distribuição geométrica, são atribuídas probabilidades de selecção para cada um dos indivíduos, de acordo com a sua ordenação. É depois seleccionado de forma aleatória um valor real entre 0 e 1, e aquele indivíduo sobre o qual incidir o valor seleccionado, de forma cumulativa, é escolhido para fazer parte da selecção.

Selecção de Progenitores

O objectivo da selecção de progenitores é seleccionar os indivíduos que se tornarão os progenitores da descendência que fará eventualmente parte da próxima geração de indivíduos, tendo sido utilizado o já descrito método de selecção por ordem.

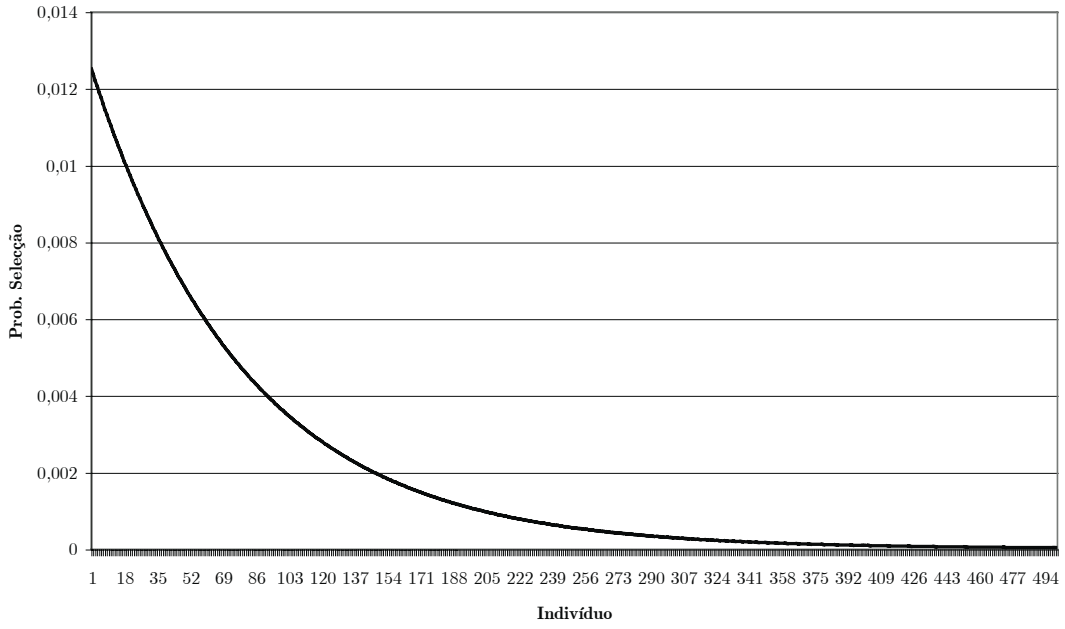


Figura 4.3: Gráfico da distribuição geométrica com as probabilidades de selecção de cada indivíduo se tornar um progenitor

Na Fig. 4.3 pode-se observar o gráfico da distribuição geométrica das probabilidades de selecção para 500 indivíduos e $p = \frac{1}{80}$, um valor também obtido de forma empírica, que origina uma curva prolongada, garantindo probabilidades de selecção ainda relevantes para elementos já próximos do fim da lista ordenada em função da

energia de conformação. Já na Fig. 4.4 observa-se a distribuição cumulativa correspondente aos valores da distribuição geométrica na Fig. 4.3.

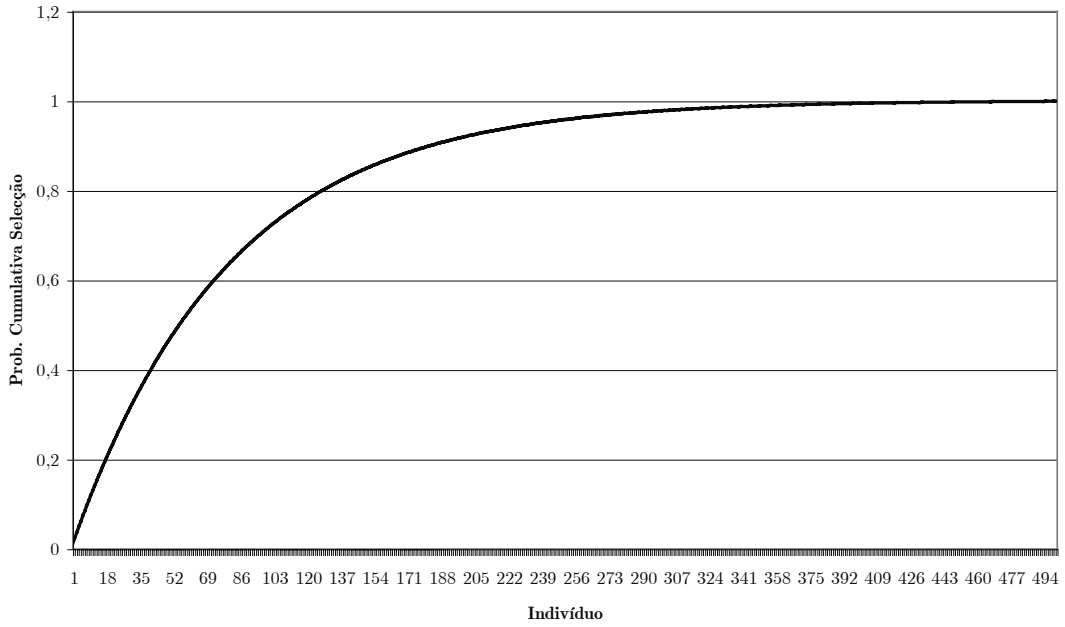


Figura 4.4: Gráfico da distribuição cumulativa das probabilidades de selecção de cada indivíduo se tornar um progenitor

Na Fig. 4.5 pode-se observar um gráfico circular com as probabilidades de os indivíduos serem seleccionados para progenitores da descendência seguinte. Os indivíduos foram reunidos em grupos de 25 para facilitar a leitura do gráfico; no entanto, a distribuição é demonstrativa do que acontece individualmente: apesar de os indivíduos mais aptos terem uma maior probabilidade de selecção, estes não detêm o "monopólio" da selecção, havendo ainda possibilidade de indivíduos menos aptos serem seleccionados. Evita-se assim que um indivíduo "superadaptado" possa ser seleccionado na grande maioria das vezes.

Selecção de Sobreviventes

Tendo-se optado por uma abordagem geracional onde existe competição entre descendência e progenitores por um lugar na próxima geração (tal como aconteceu na selecção de progenitores), foi novamente utilizado o método de selecção por ordem, com as probabilidades de cada indivíduo dadas por uma distribuição geométrica.

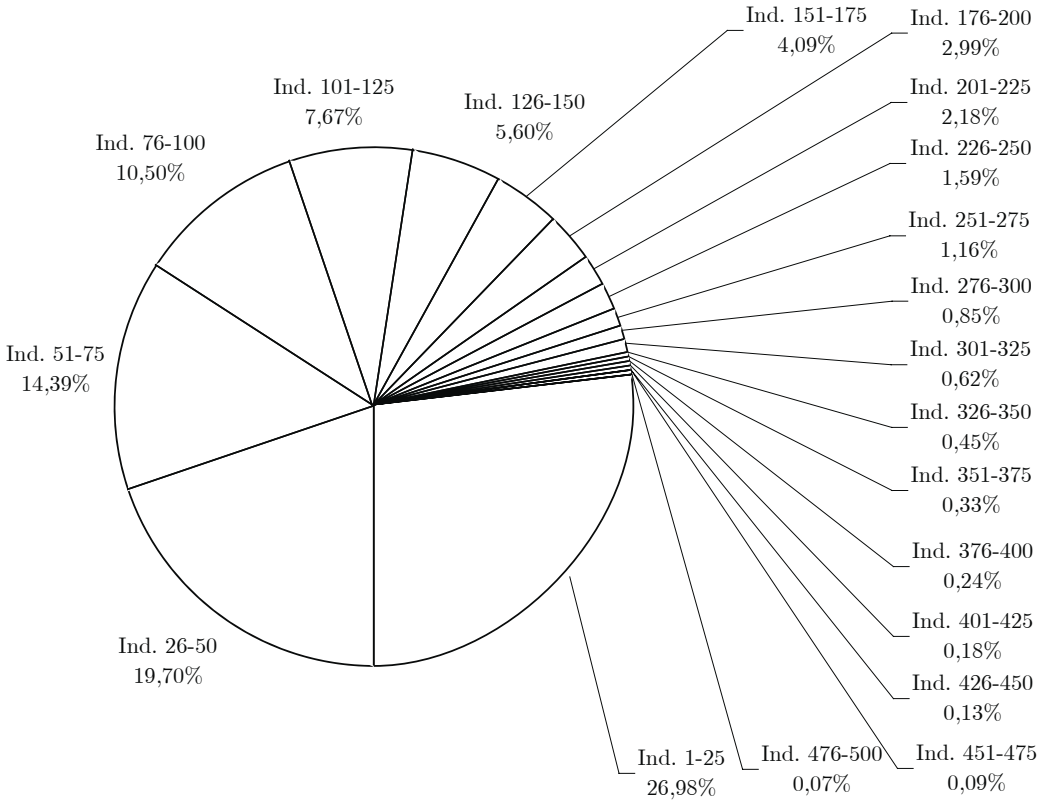


Figura 4.5: Gráfico circular da distribuição das probabilidades de selecção para progenitores em grupos de 25 indivíduos

Fundos de progenitores

Ainda sobre a selecção de progenitores, interessa referir que depois de criado um subconjunto da população constituído pelos indivíduos que foram obtidos na selecção por ordem, a recombinação não é feita de forma puramente aleatória (escolhendo-se apenas pares de indivíduos para darem origem a um novo par de indivíduos através da troca de material genético). Ao invés disso, é primeiramente seleccionado um progenitor (um dos indivíduos do conjunto com os *melhores* indivíduos) e, de seguida, também a partir do conjunto com os melhores indivíduos, é criado, de forma aleatória, um fundo — i.e., um subconjunto com uma fracção do conjunto maior —, do qual vai ser seleccionado o segundo progenitor. Esse segundo progenitor será o indivíduo mais distinto do primeiro progenitor, de acordo com a distância de Hamming,[4] de forma a criar-se um novo par de indivíduos o mais diverso possível, dadas as circunstâncias específicas do problema. Procura-se, desta forma, maximizar o operador

[4]A distância de Hamming, entre duas cadeias de caracteres de igual tamanho, é obtida contabilizando o número de posições para os quais os caracteres correspondestes são diferentes.

de recombinação, procurando-se garantir a diversidade entre os progenitores e, logo, da descendência destes.

Elite

Ainda com respeito à população, interessa referir que é utilizado o mecanismo de elite. Este mecanismo é já sobejamente conhecido em abordagens com algoritmos genéticos e o seu propósito é garantir que, durante o processo de evolução, não haja uma regressão da qualidade geral da população ou, pelo menos, dos melhores indivíduos encontrados. Garante-se assim que, na geração seguinte, há um ou mais indivíduos (tantos quanto desejado) que correspondem aos melhores encontrados nas gerações precedentes. Estes indivíduos não estão sujeitos a competição entre eles e os seus pares, bem como com os novos elementos criados na descendência, passando directamente à próxima geração.

4.2.5 Operadores de Variação

Os operadores de variação servem para a criação de novos indivíduos e subdividem-se em duas categorias: recombinação e mutação (incluindo também os casos específicos ao problema estudado, englobados sob a denominação de operadores de macromutação). São ambos aplicados neste algoritmo genético e passam a ser descritos de seguida.

Recombinação

Na implementação deste algoritmo genético foram testados vários mecanismos de recombinação, desde a recombinação com apenas um ponto de corte até à recombinação uniforme, que acabou por ser a escolhida. Esta induz uma maior diversidade da descendência gerada, possibilitando uma taxa de diversidade mais elevada, e durante mais tempo também, na população. Na recombinação uniforme, os novos indivíduos são constituídos seleccionando aleatoriamente genes de cada um dos progenitores.

Na versão utilizada, cada recombinação dá origem a dois novos indivíduos. Estes dois novos indivíduos vão sendo criados em simultâneo, sendo que sempre que um gene do primeiro progenitor for seleccionado para ser copiado para o primeiro descendente, um gene do segundo progenitor será seleccionado para ser copiado para o segundo descendente — i.e., sempre que um gene de um dos progenitores é seleccionado para

incorporar um dos descendentes, o gene na posição correspondente do outro progenitor será também seleccionado para incorporar o outro descendente. Este mecanismo pode ser observado na ver Fig. 4.6, onde está também representada a máscara, que definiu a selecção de genes de um ou de outro progenitor, obtida de forma aleatória.

À semelhança de outros operadores de variação aqui utilizados, a recombinação está sujeita a uma taxa variável de utilização entre 50 e 75% — valores comuns entre os quais varia a taxa de recombinação nos algoritmos genéticos.

Progenitor 1	*R*	*F*	*R*	*R*	*L*	*L*	*R*	*F*	*F*	*F*	*L*	*R*	*L*	*L*	*R*	*R*	*F*	*R*
Progenitor 2	**R**	**R**	**R**	**R**	**L**	**L**	**F**	**F**	**L**	**F**	**L**	**L**	**L**	**R**	**F**	**R**	**R**	**R**
Máscara	1	0	0	1	1	1	0	0	1	0	1	1	0	0	0	1	0	0
Descendente 1	*R*	**R**	**R**	*R*	*L*	*L*	**F**	**F**	*F*	**F**	*L*	*R*	**L**	**R**	**F**	*R*	**R**	**R**
Descendente 2	**R**	*F*	*R*	**R**	**L**	**L**	*R*	*F*	**L**	*F*	**L**	**L**	*L*	*L*	*R*	**R**	*F*	*R*

Figura 4.6: Operador de recombinação uniforme

Ao contrário dos outros operadores de variação, como se poderá observar de seguida, cujas probabilidades de uso crescem ao longo da execução do programa ou das várias etapas da sua execução, a probabilidade de recombinação vai decrescendo. Tal acontece porque a partir do momento em que a população converge (aproximando-se da homogeneidade), a probabilidade de gerar novos indivíduos através de recombinação diminui drasticamente. Assim, passa-se a diminuir a utilização do mecanismo de recombinação e passa-se a dar maior destaque à variação por mutação e macro-mutação.

Mutação

O mecanismo de mutação utilizado consiste na alteração de valores de cada um dos elementos do genótipo (os genes), fazendo variar os seus valores (alelos). Se a posição actual assumir, por exemplo, o valor L (esquerda), tem iguais probabilidades de se tornar R (direita) ou F (frente). Foi tomada a opção de evitar que uma mutação produza um resultado idêntico ao actual. Assim, por exemplo, neste caso não é considerada uma transição de L para L.

Por opção, cada um dos genes tem uma pequena probabilidade de ser mutado, inversamente proporcional ao comprimento C do genótipo: $\frac{1}{C}$. Quando tal se verifica,

o operador cessa a sua execução — i.e., no máximo, apenas um gene é mutado em cada aplicação do operador a um indivíduo. A razão desta opção deve-se ao facto de as várias sequências de teste terem tamanhos bastante díspares, variando entre 18 e 100 *aminoácidos*, o que poderia levar a que, em indivíduos com genótipos maiores, o operador de mutação fosse mais vezes aplicado que em indivíduos com genótipos menores. Optou-se também por usar probabilidades de mutação relativamente baixas, variando entre 1 e 2%, para evitar que este mecanismo seja demasiado destrutivo, permitindo ao mesmo tempo que o indivíduo sofra alterações que, de facto, o façam diferenciar-se da sua forma inicial (ver Fig. 4.7).

Desc. Original 1	R	R	R	*R*	L	L	F	F	F	F	L	R	L	R	F	R	R	R
Desc. Original 2	R	F	R	R	L	L	R	F	L	F	L	L	**L**	L	R	R	F	R
Desc. Mutado 1	R	R	R	*F*	L	L	F	F	F	F	L	R	L	R	F	R	R	R
Desc. Mutado 2	R	F	R	R	L	L	R	F	L	F	L	L	**R**	L	R	R	F	R

Figura 4.7: Operador de mutação

Interessa referir que o mecanismo de mutação, tal como está estruturado o algoritmo genético do programa, só é aplicado se nenhum dos mecanismos de macromutação — que serão abordados de seguida — for aplicado.

Macromutação

Para além da mutação mais convencional, optou-se pela utilização de técnicas específicas de mutação adequadas a este problema, que têm o nome genérico de macromutações. Os operadores de macromutação, de uma forma geral, trabalham sobre excertos do genótipo. Dos macromutadores utilizados, uns procuram padrões no genótipo dos indivíduos — as sequências com as posições relativas de cada aminoácido — e substituem esses padrões por outros predefinidos,[5] outros apenas alteram a ordem ou trocam genes de uma forma agrupada. Os operadores de macromutação, à semelhança do que acontece com o operador de mutação, são executados com uma probabilidade variável, neste caso entre 10 e 20% — limites estes que são flexíveis, como se poderá observar no *Capítulo 5*. São utilizados estes valores (altos) devido ao facto de os macromutadores só serem aplicados em situações bastante específicas,

[5]Para alguns exemplos de macromutadores com substituição de padrões, veja-se [RCMJJ04].

quando determinado padrão é encontrado, pelo que é necessário que haja a uma probabilidade significativa para a sua real aplicação.

Seguem-se os vários operadores de macromutação utilizados,[6] começando por aqueles que realizam busca de padrões, substituindo esses padrões por outros:

- **Desdobragem** — este operador (ver Fig. 4.8) procura, no genótipo, os padrões *RLLR*, *LRRF*, *RLLR* e *RLLF*, e, caso os encontre, troca-os pelos padrões *FFFF*, *FFFR*, *FFFF* e *FFFL*, respectivamente.

- **Dobragem** — este operador (ver Fig. 4.9) procura o padrão *FFFF* no genótipo e, caso o encontre, troca-o pelos padrões *RLLR* ou *RLLR*;

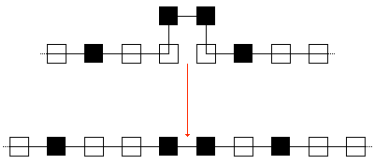


Figura 4.8: Operador de macromutação desdobragem

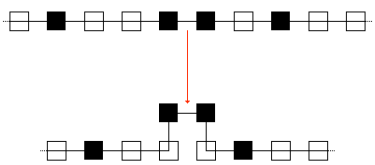


Figura 4.9: Operador de macromutação dobragem

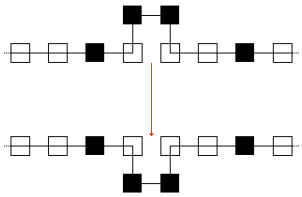


Figura 4.10: Operador de macromutação manivela

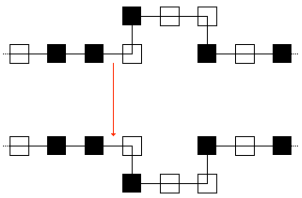


Figura 4.11: Operador de macromutação rotação

- **Manivela** — este operador (ver Fig. 4.10) procura o padrão *LRRL* no genótipo e, caso o encontre, troca-o pelo padrão *RLLR*, e *vice versa*;

- **Rotação** — este operador (ver Fig. 4.11) procura os padrões *LRFRF*, *LRFRL*, *RLFLF* e *RLFLR*, e troca-os por *RLFLF*, *RLFLR*, *LRFRF*, *LRFRL*, respectivamente;

- **Serpenteação** — este operador (ver Fig. 4.12) procura os padrões *FLRRL*, *FLRRF*, *FRLLR* e *FRLLF*, e fá-los avançar uma posição na ordem do genótipo,

[6] Os nomes para os vários operadores de macromutação podem não ser consensuais, visto serem usados predominantemente em inglês. Para contornar essa situação, seguem-se os nomes em português e em inglês: manivela — *crank*; dobragem — *fold*; inserção — *insert*; inversão — *invert*; vincagem — *kink*; rotação — *rotate*; baralhação — *scramble*; serpenteação — *snake*; troca — *swap*; troca-sequência — *swap-sequence*; e desdobragem — *unfold*.

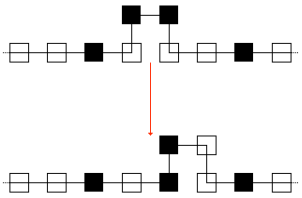


Figura 4.12: Operador de macromutação serpenteação

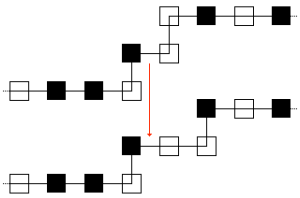


Figura 4.13: Operador de macromutação vincagem

substituindo-os pelos padrões *LRRLF*, *LRRLR*, *RLLRF* e *RLLRL*, respectivamente;

- **Vincagem** — este operador (ver Fig. 4.13) procura os padrões *LRF* ou *RLF* e troca-os por *FLR* ou *FRL*, respectivamente.

Por outro lado, também são utilizados operadores de macromutação, aplicados independentemente da ocorrência ou não de padrões, que essencialmente realizam permutas de segmentos do genótipo de genes, alterando a sua ordem ou posição dentro do genótipo. Estes macromutadores são:

- **Baralhação** — este operador (ver Fig. 4.14) selecciona um excerto do genótipo e troca a ordem dos genes contidos nesse excerto;

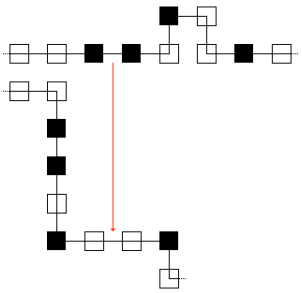


Figura 4.14: Operador de macromutação baralhação

- **Inserção** — este operador (ver Fig. 4.15) selecciona aleatoriamente um gene, retira-o e insere-o noutra posição também escolhida aleatoriamente, deslocando os genes compreendidos entre a nova posição e a anterior;

- **Inversão** — este operador (ver Fig. 4.16) selecciona aleatoriamente uma secção do gene e inverte a ordem pela qual os genes se encontram nessa secção;

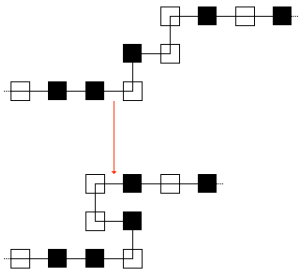


Figura 4.15: Operador de macromutação inserção

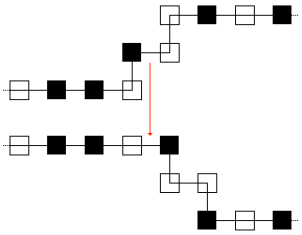


Figura 4.16: Operador de macromutação inversão

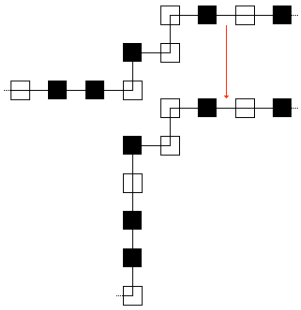


Figura 4.17: Operador de macromutação troca

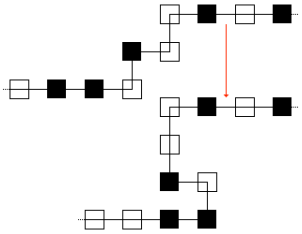


Figura 4.18: Operador de macromutação troca-sequência

- **Troca** — este operador (ver Fig. 4.17) selecciona aleatoriamente dois genes e troca as suas posições;
- **Troca-sequência** — este operador (ver Fig. 4.18) selecciona aleatoriamente duas secções do genótipo, não sobrepostas e de igual tamanho, e troca as suas posições.

Apesar de haver alguns macromutadores que produzem resultados semelhantes, como o *troca* e o *troca-sequência*, ou resultados opostos, como o *dobragem* e o *desdobragem*, optou-se por manter tal diversidade de operadores essencialmente por dois motivos: nunca dois macromutadores são utilizados em simultâneo, não havendo o risco de um reverter o efeito do outro; e não se quis fazer uma avaliação prematura da eficácia de cada um deles, mantendo-se a utilização de todos. Para além disso, poder-se-ia dar o caso de em iterações sucessivas do algoritmo, haver *combinações* de macromutações que pudessem dar origem a resultados favoráveis (se bem que tal, à partida, seja difícil de aferir).

A utilização de operadores de macromutação justifica-se essencialmente por dois motivos. Em primeiro lugar, está-se a manipular grupos de genes ao mesmo tempo

(no mínimo dois genes), geralmente em posições vizinhas, levando a que a alteração tenha maior expressão no fenótipo do indivíduo. Este motivo é válido tanto para os macromutadores com busca de padrões como para aqueles sem busca de padrões. Em segundo lugar, com os macromutadores com busca de padrões (e sua substituição), está-se a trabalhar com conjuntos de genes que originam conformações localizadas (que podem ocorrer com bastante frequência) que, dependendo do sítio onde se encontram, podem contribuir para um fenótipo melhor ou pior adaptado, com a correspondente energia mínima de conformação.

Taxas Dinâmicas de Variação

Foi adoptada a técnica de taxas de variação variáveis de forma a injectar na população, de tempos a tempos e sempre que esta começa a estagnar, alguma diversidade. Assim, quando a população começa a ficar homogeneizada, o peso da recombinação na criação de novos indivíduos vai decrescendo e os pesos da mutação e da macromutação vão crescendo — como foi já referido na descrição de cada um destes operadores. Posto de outra forma, quando a população é heterogénea, o principal impulsionador da criação de novos indivíduos e da evolução é a recombinação; quando a população é homogénea, a recombinação perde o seu significado e o impulsionador, criador de diversidade, passa a ser a mutação (englobando os já anteriormente referidos operadores de macromutação).

Nas Figs. 4.19, 4.20 e 4.21 pode ser observada a variação das taxas de mutação, recombinação e macromutação, respectivamente; na Fig. 4.22 pode-se observar a sobreposição das probabilidades para os vários operadores de variação. São apresentados os gráficos para 1000 gerações; no entanto, este valor não é predeterminado ou fixo, já que, tal como se encontra implementado o algoritmo, as alterações aos valores base iniciais são efectuadas até que haja evolução na população — i.e., quando é encontrado um indivíduo com uma energia de conformação menor que a actual — ou indefinidamente até se chegar ao número máximo de gerações sem qualquer evolução na população.

4.2.6 Reparação

Um grande factor de diferenciação no algoritmo genético aplicado ao problema estudado é a utilização de um mecanismo de reparação, evitando que indivíduos com potencial (e talvez próximos de uma melhor solução) sejam descartados devido ao

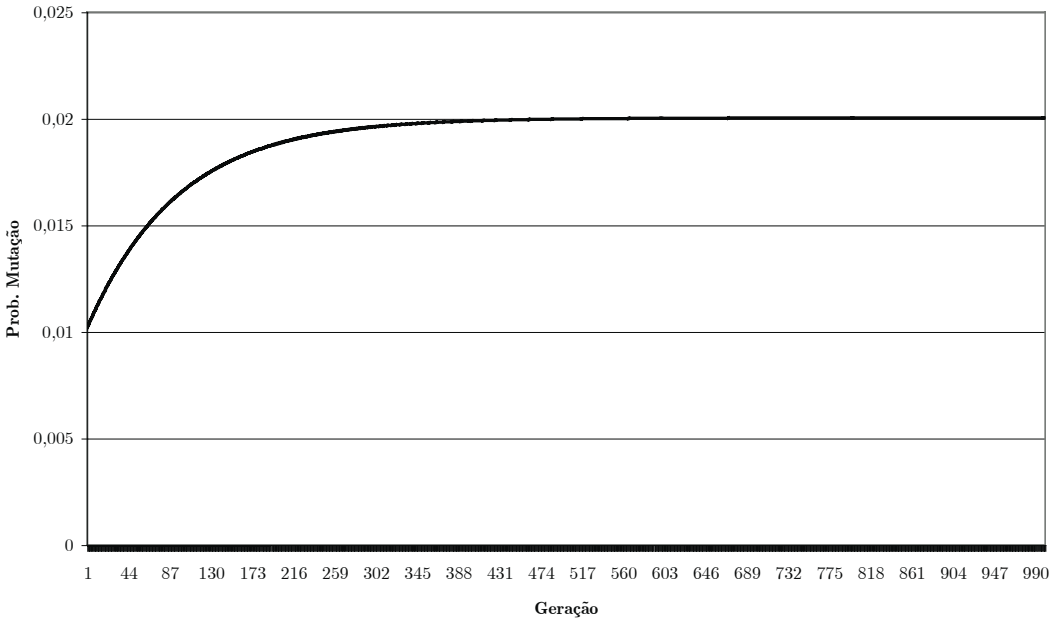


Figura 4.19: Gráfico da probabilidade variável de mutação ao longo das gerações de uma execução

incumprimento de uma das regras de definem uma boa solução candidata. Assim, procura-se reparar o indivíduo, tentando ao mesmo tempo que mantenha as características que lhe atribuiram destaque aquando da aplicação da função de avaliação — o número de contactos H–H, em aminoácidos não adjacentes.

Para além disto, já que a função de reparação, por questões de eficiência,[7] só tenta um número limitado de reparações, é também permitido que indivíduos, mesmo que sejam inválidos, possam passar à geração seguinte, se bem que penalizados pelos motivos que os levam a ser caracterizados como inválidos.

O mecanismo de reparação é apresentado com maior detalhe mais adiante na *Secção 4.3*, sendo descritos o seu funcionamento e aplicabilidade.

[7]Há dois aspectos a ter em conta em termos de eficiência. Em primeiro lugar, os recursos de memória da *Java Virtual Machine*, já que é um mecanismo que funciona forma iterativa, podendo exaurir a memória disponibilizada. Em segundo lugar, o facto de o tempo (de processador) despendido na reparação não compensar o benefício obtido, havendo mesmo a possibilidade de se entrar em ciclo, em que uma reparação pode dar origem a uma nova conformação inválida e assim sucessivamente.

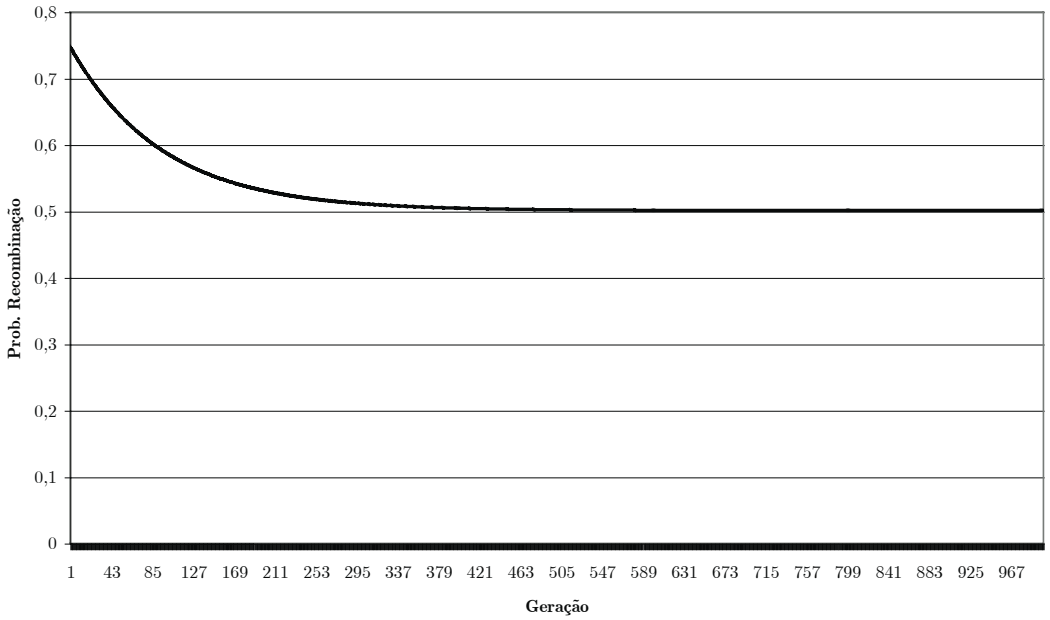


Figura 4.20: Gráfico da probabilidade variável de recombinação ao longo das gerações de uma execução

4.2.7 Condição de Paragem

A condição de paragem depende em muito do problema em estudo. Sendo este caso específico um problema de optimização — mesmo estando-se a trabalhar com sequências de teste padrão, com resultados conhecidos *a priori* para efeitos de comparação — sem conhecimento da solução óptima *a priori*, a condição de paragem tem de ser obrigatoriamente uma de entre:

1. uma solução encontrada satisfazer os critérios mínimos;
2. inspecção "manual";
3. a função do tempo de execução;
4. o número de gerações;
5. a estagnação da qualidade dos indivíduos que constituem a população.

Neste algoritmo genético a condição de paragem é uma conjunção das duas últimas possibilidades. Garante-se assim, à partida, que há um número mínimo de gerações a criar e, a partir daí, a condição de paragem passa a ser a estagnação dos indivíduos da população.

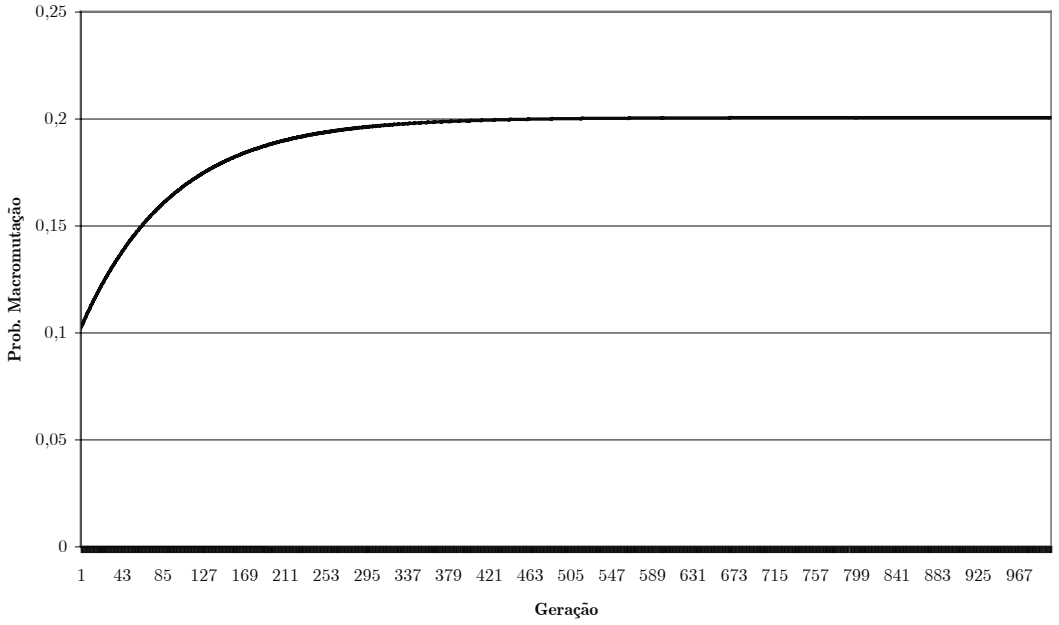


Figura 4.21: Gráfico da probabilidade variável de macromutação ao longo das gerações de uma execução

4.2.8 Parâmetros do Algoritmo Genético

Após todo um processo de evolução da implementação do programa, com avanços e recuos, a versão do final do algoritmo genético é configurada através dos seguintes parâmetros:

- **proteína** — a cadeia com a caracterização dos aminoácidos em termos de serem hidrofóbicos ou polares;
- **tamanho da população** — número de indivíduos que constitui a população em cada uma das iterações;
- **tamanho do fundo de progenitores** — número de indivíduos que constituirão o fundo (*"pool"*) para selecção dos progenitores da geração seguinte;
- **tamanho da elite** — número de indivíduos constituintes da elite (zero, se não se usar uma abordagem elitista, maior que zero se for usada uma abordagem elitista; tipicamente utilizam-se valores pequenos, na casa da uma ou das duas unidades);
- **número de saída** — número mínimo de gerações após o qual o programa

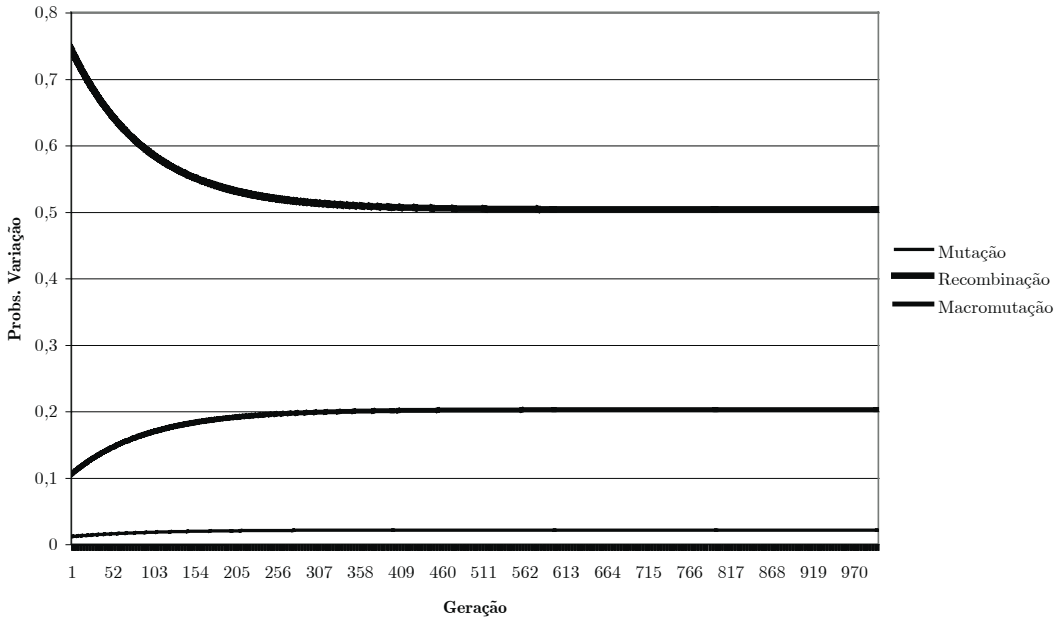


Figura 4.22: Gráfico das probabilidades variáveis de variação sobrepostas ao longo das gerações de uma execução

pode terminar (mas só depois de um determinado número de gerações ter sido atingido *a priori*);

- **valor da penalidade** — número de pontos de penalidade atribuídos aos indivíduos por cada atributo que os torne inválidos, o que se traduz em sobreposições;

- **tamanho da descendência** — número de indivíduos da descendência (regra geral, calculado em função do tamanho da população);

- **tamanho da selecção de progenitores** — número de indivíduos seleccionados para dar origem à descendência;

- **número mínimo de gerações** — número mínimo de gerações que têm de existir antes que o programa possa terminar (após atingido este valor, ainda se tem de verificar o *número de saída*);

- **patamar de reparação** — percentagem sobre o valor da qualidade do melhor indivíduo de cada geração, acima do qual um indívíduo inválido deve ser reparado;

- **patamar de número de gerações sem evolução** — valor utilizado para controlo do número de gerações sem evolução nos indivíduos e após o qual se começa a alterar os valores de mutação, recombinação e macromutação, para aumentar a diversidade da população (ou, pelo menos, da descendência);

- **probabilidade inicial de mutação** — valor inicial da probabilidade dinâmica de mutação;

- **probabilidade final de mutação** — valor final da probabilidade de dinâmica mutação;

- **probabilidade inicial de macromutação** — valor inicial da probabilidade de dinâmica macromutação;

- **probabilidade final de macromutação** — valor final da probabilidade de dinâmica macromutação;

- **probabilidade inicial de recombinação** — valor inicial da probabilidade de dinâmica recombinação;

- **probabilidade final de recombinação** — valor final da probabilidade de dinâmica recombinação;

- **probabilidade da distribuição geométrica** — valor da probabilidade de distribuição geométrica, utilizada na selecção de indivíduos;

- **probabilidade da distribuição geométrica da mutação** — valor da probabilidade de distribuição geométrica, utilizada na variação da probabilidade de mutação dos indivíduos;

- **probabilidade da distribuição geométrica da recombinação** — valor da probabilidade de distribuição geométrica, utilizada na variação da probabilidade de recombinação dos indivíduos;

- **probabilidade da distribuição geométrica da macromutação** — valor da probabilidade de distribuição geométrica, utilizada na variação da probabilidade de macromutação dos indivíduos.

4.3 Mecanismo de Reparação

O mecanismo de reparação é aplicado a todos os indivíduos; no entanto, só se estes estiverem efectivamente inválidos é que sofrem alguma tentativa de reparação. Tal deve-se ao facto de o primeiro passo em cada iteração do mecanismo de reparação ser verificar a validade do indivíduo, para que se possa determinar se se prossegue com a sua execução (continuar a reparar o indivíduo, passando-se a uma nova iteração) ou se o indivíduo já se encontra reparado, terminando-se o processo. É assim escusado fazer essa verificação prévia antes de se aplicar o mecanismo de reparação. O mecanismo, tal como se encontra implementado, altera o próprio indivíduo (ver Alg. 11). Passa-se a descrever o mecanismo de reparação:

1. Verifica-se, inicialmente, se o indivíduo necessita de ser reparado (um indivíduo necessita de reparação quando é inválido — i.e., possui mais que um aminoácido a ocupar a mesma posição):
   (a) o mecanismo começa por obter a estrutura bidimensional do indivíduo, transformando as direcções codificadas no genótipo em coordenadas;
   (b) de seguida, vai criando uma lista com todas as coordenadas e, à medida que as vai adicionando, verifica se alguma delas se encontra repetida;
   (c) a partir do momento em que encontra uma coordenada repetida — i.e., há dois aminoácidos com as mesmas coordenadas, e logo uma sobreposição em determinada posição —, o indivíduo é dado como inválido e passa ao processo de reparação;
2. Em caso de ser necessária a reparação, procura-se o primeiro ponto de intersecção do "caminho" do indivíduo (começando aleatoriamente pela cauda ou pela cabeça) e altera-se o valor do alelo correspondente ao aminoácido imediatamente anterior à posição sobreposta. Com maior pormenor:
   (a) procura-se a primeira posição sobreposta (no fenótipo) e identifica-se (no genótipo) qual o gene que originou-se essa sobreposição e altera-se o seu valor — e.g., de L para F ou R;
   (b) de seguida, verifica-se se a nova posição gerada não está também sobreposta:
      i. se sim, recua-se um gene na sequência e repete-se este passo até estar feita a reparação — note-se que aqui apenas se faz uma reparação

localizada, podendo dar-se o caso desta reparação ter repercussões em outras partes do indivíduo;

ii. se não, passa-se ao passo seguinte;

3. Repete-se o processo de forma a verificar se o indivíduo ficou reparado, sem qualquer sobreposição — volta-se a testar *todo* o indivíduo (passo 1), após as reparações localizadas que aconteceram no passo anterior.

```
Método Reparar
Entrada: indivíduo
Saída: indivíduo'
se não existem sobreposições no fenótipo ou número máximo de tentativas
alcançado então
    retornar indivíduo
senão
    seleccionar aleatoriamente a ponta por onde começar;
    procurar o primeiro aminoácido numa posição sobreposta;
    enquanto sobreposição local fazer
        alterar gene responsável pela sobreposição → indivíduo';
        se o novo valor não criar uma nova sobreposição com o mesmo
        aminoácido então
            reparar indivíduo';
        senão
            recuar um gene;
        fim
    fim
    reparar indivíduo';
fim
```

Algoritmo 11: Algoritmo do mecanismo de reparação

Note-se que é necessária a definição de um número máximo de tentativas de reparação, devido ao facto de que as próprias tentativas de reparação, só por si, poderem causar novas sobreposições noutros locais, ou mesmo dar-se o caso de se entrar em ciclo infinito com as reparações, passando-se de uma sobreposição a outra e voltando à inicial. Actualmente este valor está definido empiricamente, tendo sido usado um número que evite que a *Java Virtual Machine* aborte os processos devido à escassez de recursos.[8] Quanto maior for a capacidade do(s) computador(es) onde o programa é

[8] Ao utilizar um método que funciona de forma recursiva — e isto é válido tanto para *Java* como para qualquer outra linguagem de programação — se não for garantido que o elemento recursivo termina, evitando um ciclo infinito e antes que a memória disponível para a aplicação se esgote, o processo acabará inevitavelmente por ser abortado como medida de preservar a estabilidade do sistema.

executado, maior poderá ser o número de tentativas disponibilizadas. Por outro lado, o processo de reparação é um dos pontos mais consumidores de tempo do programa, pelo que foi necessário chegar a um compromisso encontrado empiricamente.

A título de exemplo, nas Figs. 4.23 e 4.24 podem ser observadas as alterações observadas no genótipo e no fenótipo, respectivamente, durante o processo de reparação de um indivíduo classificado como inválido. Utilizou-se a sequência de teste padrão número um e começou-se com um indivíduo completamente "enrolado". Especificamente, na Fig. 4.23 pode ser observada a aleatoriedade na escolha da cabeça ou da cauda do genótipo para se percorrer a sequência de genes e se proceder à reparação do indivíduo em cada uma das iterações. Note-se, também, que a reparação apresentada é apenas uma das possíveis, já que, para além de a escolha da ponta do genótipo por onde começar ser feita de forma aleatória, o mesmo acontece com o novo valor (direcção) que o gene "prevaricador" irá assumir.

```
0 LLLLLLLLLLLLLLLLL
1 LLFLLLLLLLLLLLLLL
2 LLFLLLLLLLLLLLLLF
3 LLFLRLLLLLLLLLLLF
4 LLFLRLLFLLLLLLLLF
5 LLFLRFLFLLLLLLLLF
6 LLFLRFLFLLRLLLLLF
7 LLFLRFLFLRRLLLLLF
8 LLFLRFLFLRRLLFLLF
9 LLFLRFLFLRRLLFLRLF
```

Figura 4.23: Alteração do genótipo do indivíduo ao longo das várias iterações da reparação

4.4 Programa

O programa foi implementado com recurso à linguagem de programação *Java* (JDK 6), desenvolvido pela *Sun Microsystems*. Esta escolha deveu-se a uma série de aspectos de ordem prática, entre os quais se destacam:

- o problema poder ser bem definido em termos de várias partes distintas, que podem ser transpostas para uma linguagem de programação orientada objectos, onde podem ser identificados características e comportamentos dos vários elementos do programa;

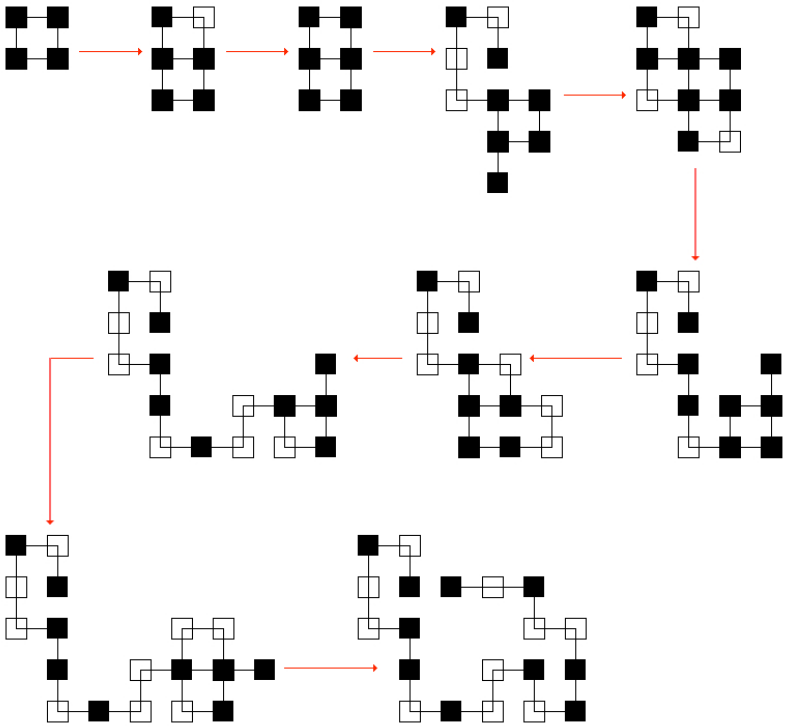


Figura 4.24: Alteração do fenótipo do indivíduo ao longo das várias iterações da reparação

- a familiaridade do autor com a linguagem;
- a portabilidade da linguagem, facilitando a transição entre o equipamento do autor e o equipamento onde foram realizados os testes aqui apresentados — o que correspondeu a sistemas operativos e a *hardware* distintos.

Segue-se uma descrição do programa, desenvolvido em *Java*, para suporte ao algoritmo genético aqui proposto, apresentando-se o fluxograma de execução do programa principal (ver Fig. 4.25) e as diversas classes, bem como os respectivos métodos (ver Fig. 4.26).

As várias classes definem aspectos distintos do programa (e, consequentemente, do algoritmo genético) e passam a ser descritas de seguida:

- **AminoAcid** — esta classe serve essencialmente para a ser utilizada na representação do fenótipo, guardando o tipo de cada aminoácido, bem como a sua posição (aqui já num sistema eixos cartesianos);
- **DrawArea** — componente gráfico utilizado no *ModelDrawer* para desenhar o fenótipo;

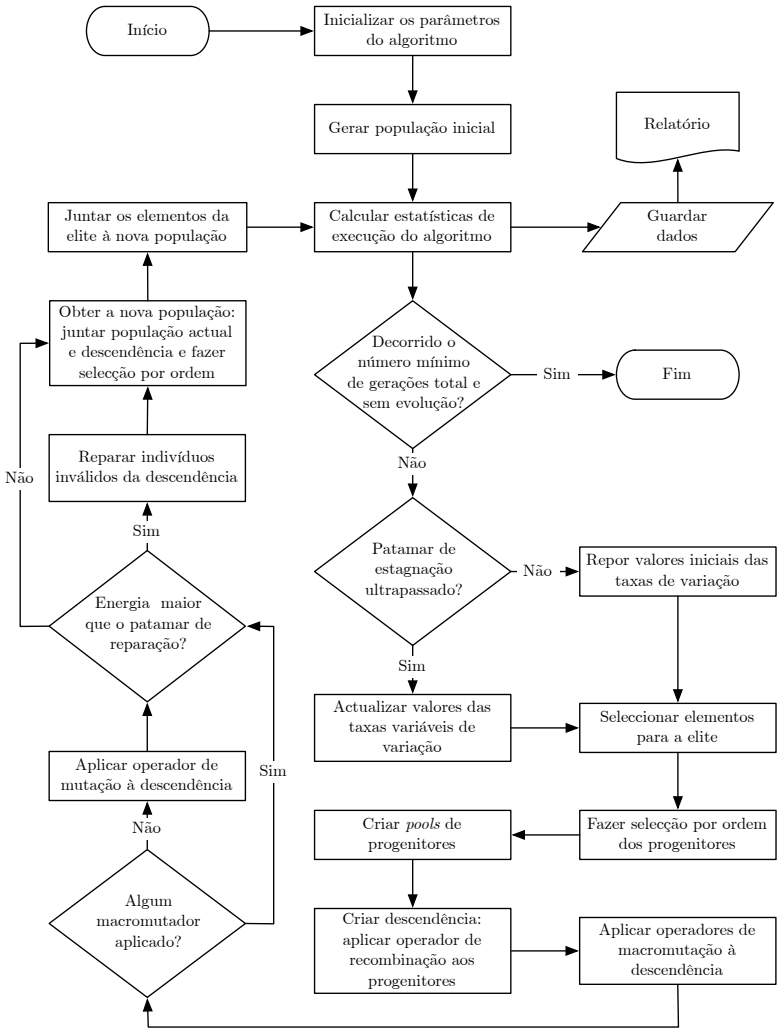


Figura 4.25: Fluxograma de execução do programa

- **GlobalReporter** — aplicação auxiliar para manipular o resultado das execuções para uma dada sequência de aminoácidos e aglutiná-los num só ficheiro global;

- **HPModel2D** — esta é a classe principal do programa e é aqui que se define a estrutura base do algoritmo genético utilizado, definindo-se, por exemplo, os parâmetros utilizados, forma de obter a descendência e de aplicar os operadores de variação, bem como o mecanismo de reparação;

- **HPModel2DConstants** — aqui encontram-se as constantes comuns a várias classes do programa, de forma a garantir a coerência dos dados por elas representados;

- **HPModel2DViewer** — interface gráfico no qual se pode inserir uma sequência de aminoácidos e o genótipo de um indivíduo e obter o seu fenótipo;
- **Individual** — representa cada uma das soluções, guardando o genótipo de cada indivíduo e permitindo, entre outras operações, a verificação da validade do indivíduo, a recombinação (com outro indivíduo), a sua mutação e várias macromutações, bem como o mecanismo de reparação;
- **IndividualEnergy** — serve para guardar a energia de um dado indivíduo, especialmente útil para ordenamentos da população em função da energia dos indivíduos, evitando-se a constante (re)avaliação do indivíduo;
- **Model** — serve para representar as características gerais ao modelo, como a proteína (ou, melhor dizendo, a sequência das características hidrofóbicas e polares dos aminoácidos) e o valor da penalidade, e permite também avaliar os indivíduos, ordená-los, seleccioná-los e obter o seu fenótipo;
- **ModelDrawer** — componente gráfico para mostrar o fenótipo de um ou mais indivíduos, bem como a sequência, genótipo e pontos de energia;
- **ModelGUI** — interface gráfico para interagir com o programa, permitindo passar os parâmetros e argumentos usando-se o ambiente gráfico com janelas, em vez da linha de comando.

Note-se que na Fig.4.26 encontra-se a classe *GlobalReporter* de forma isolada. Tal acontece porque esta apenas serve para aglutinar num só ficheiro, para posterior análise, os dados de todos os ficheiros resultantes de cada uma das execuções do programa para cada uma das sequências, tendo apenas acesso aos ficheiros resultantes das execuções do programa. Também as classes *ModelGUI* e *HPModel2DViewer*, se bem que não isoladas, têm um papel secundário, facilitando apenas o acesso ao programa, quando se pretende configurar o algoritmo e obter os resultados através do ambiente gráfico.

Por outro lado, o interface com programa, tal como executado no *cluster* para obter os resultados aqui apresentados, foi feita utilizando apenas a linha de comando, programas de *batch* e ficheiros (para guardar os dados de execução).

Devido à sua extensão e detalhes técnicos, mais pormenores sobre cada uma das classes, tais como uma descrição mais exaustiva, atributos e métodos, podem ser encontrados na documentação do programa, presente em anexo, no CD.

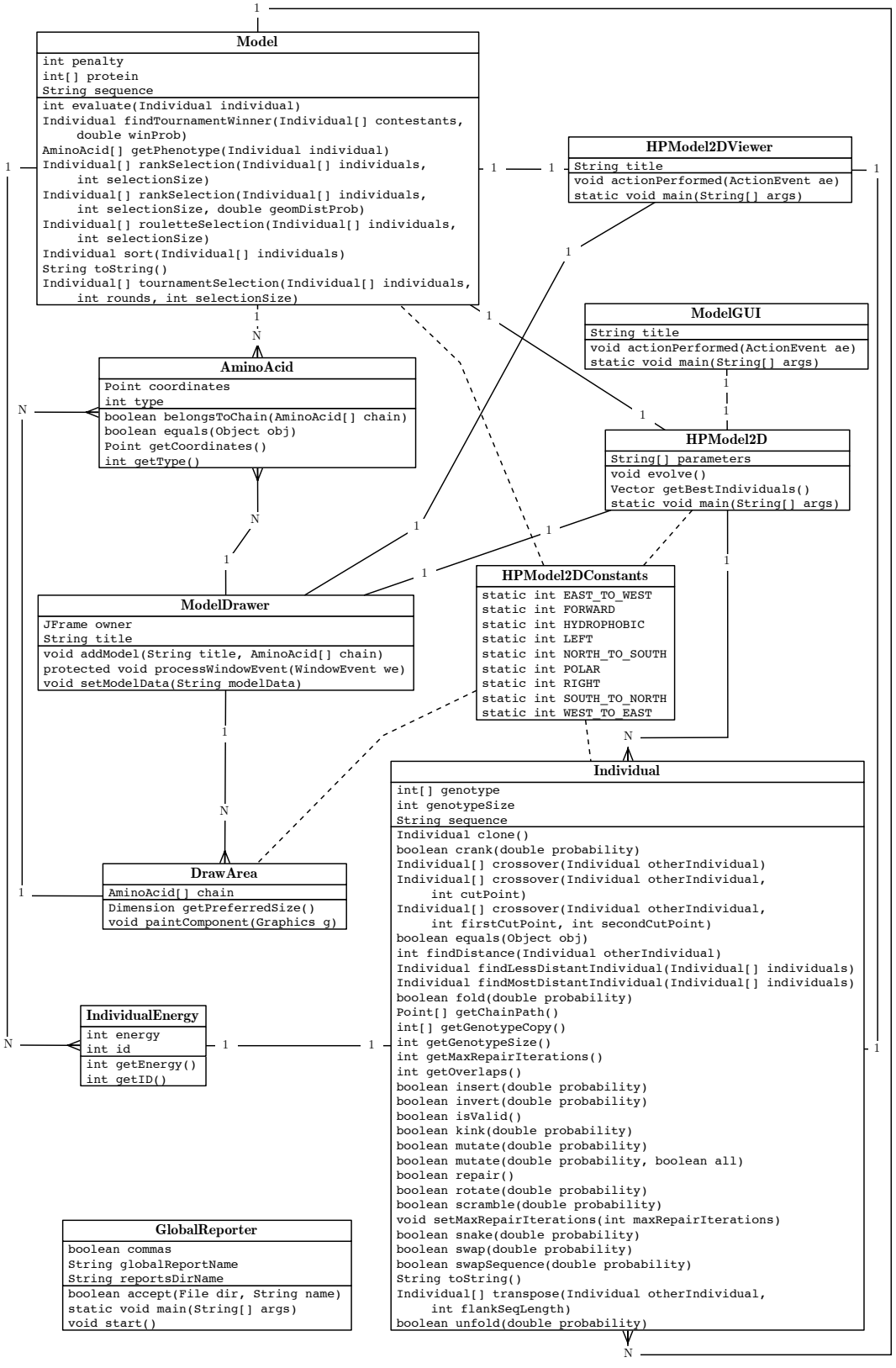


Figura 4.26: Diagrama de classes do programa

Capítulo 5

Resultados

Neste capítulo são analisados os resultados obtidos com o método proposto, sendo mostradas as vantagens da utilização de um mecanismo de reparação de indivíduos inválidos, da utilização de operadores de mutação específicos ao problema estudado — os operadores de macromutação — e, também, as implicações da utilização de taxas variáveis na aplicação dos operadores de variação (sejam estes de recombinação ou de mutação).

5.1 Condições de Execução

Os resultados sobre o desempenho do algoritmo proposto para aplicação ao modelo HP foram encontrados aplicando-o às "sequências *Tortilla*", já descritas no *Capítulo 3*, que consistem em sequências de caracteres, definidas no alfabeto $\{H,P\}$, que descrevem as propriedades hidrofóbicas e polares dos aminoácidos.

Para testar o algoritmo foram realizadas 30 execuções, envolvendo os quatro conjuntos de parâmetros diferentes e as várias sequências de teste padrão, utilizando-se o mecanismo de reparação. Para além destas execuções, foram utilizados mais dois conjuntos de execuções, com o primeiro conjunto de parâmetros, para as sequências de teste padrão: um sem a utilização do mecanismo de reparação de indivíduos e sem aceitar indivíduos inválidos e outro também sem aplicar o mecanismo de reparação, mas aceitando indivíduos inválidos. Estes dois outros conjuntos serviram para efeitos de comparação e obtenção de conclusões sobre o mecanismo de reparação. No primeiro conjunto de execuções foram testadas todas as 15 sequências *Tortilla*; nos conjuntos seguintes foram testadas apenas as sequências cujos resultados obtidos (nas execuções do primeiro conjunto de parâmetros e também no segundo) não igualaram os melhores valores conhecidos até à data, procurando-se

melhorar esses resultados; concorrentemente, o primeiro conjunto de testes voltou a ser executado sem reparação, com e sem aceitação de indivíduos inválidos. Resulta assim que, para os resultados apresentados nesta dissertação, foram realizadas $(15 \times 30 + 6 \times 30 + 5 \times 30 + 5 \times 30) + (15 \times 30) + (15 \times 30) = 1830$ execuções.

Os parâmetros do algoritmo genético para os conjuntos de execuções foram em tudo idênticos, à excepção dos seguintes: *tamanho da população*, *número mínimo de gerações*, *número de saída*,[1] e *probabilidade de macromutação*. Na probabilidade de macromutação estão incluídos dois parâmetros, já que esta probabilidade é dinâmica, sendo necessário definir os valores inicial e final, entre os quais varia a probabilidade. Tais parâmetros podem ser observados na Tab. 5.1, onde se encontram destacados, em letras carregadas, os valores que foram alterados para as execuções com os diversos conjuntos de parâmetros. O número restrito de parâmetros variáveis justifica-se, em parte, por ser impraticável estudar diferentes valores para todos os parâmetros (mais ainda, dada a gama de valores possíveis). Assim, houve também parâmetros que assumiram um valor *fixo*, ou apenas com algumas variações no início dos testes ao algoritmo, numa fase de afinação.

A escolha dos parâmetros a testar com diferentes valores nos diversos conjuntos de execuções (ver Tabs. 5.3, 5.6, 5.7 e 5.8) foi baseada no seguinte:

- com o aumento do tamanho da população e do número de gerações (incluindo-se aqui o número mínimo de gerações e o número de saída), aumenta-se o espaço de procura de soluções (sendo esta uma das soluções mais usadas para este efeito);
- com o aumento, ou diminuição, da probabilidade de macromutação, dá-se um maior azo a que mutações específicas ao problema em estudo sejam aplicadas com uma maior taxa de sucesso (e dá-se a oportunidade para estudar a eficiência dos mesmos operadores).

Os outros parâmetros ficaram fixos por terem sido estabelecidos durante a construção do próprio programa (e ao longo das suas várias versões), de forma empírica, apresentando resultados satisfatórios. Assim:

- o fundo de progenitores teve um valor pequeno, inferior a 10%, por se ter verificado que esta amostra dos melhores indivíduos seria suficiente para a recombinação e preservaria os melhores genes;

[1] Número de gerações sem qualquer evolução após o qual a execução do algoritmo termina; é preciso, no entanto, que o número mínimo de gerações tenha já sido atingido.

Tabela 5.1: Parâmetros do algoritmo genético

Parâmetro	**1.º Conj.**	**2.º Conj.**	**3.º Conj.**	**4.º Conj.**
Tamanho da População	**500**	**1000**	**1500**	**2000**
Tamanho da *pool*	7,5% Pop.	7,5% Pop.	7,5% Pop.	7,5% Pop.
Tamanho da Elite	2	2	2	2
Número de Saída	**250**	**500**	**750**	**1000**
Valor da Penalização	2	2	2	2
Tamanho da Descendência	135% Pop.	135% Pop.	135% Pop.	135% Pop.
Tamanho da Selecção de Progenitores	65% Pop.	65% Pop.	65% Pop.	65% Pop.
Número Mínimo de Gerações	**1000**	**2000**	**3000**	**4000**
Patamar de Reparação	62,5%	62,5%	62,5%	62,5%
Patamar N.º Gerações sem Evolução	50%	50%	50%	50%
Prob. Inicial de Mutação	1%	1%	1%	1%
Prob. Final de Mutação	2%	2%	2%	2%
Prob. Inicial de Macromutação	**10%**	**15%**	**20%**	**15%**
Prob. Final de Macromutação	**20%**	**30%**	**40%**	**30%**
Prob. Inicial de Recombinação	75%	75%	75%	75%
Prob. Final de Recombinação	50%	50%	50%	50%
Prob. Distribuição Geométrica	1,25%	1,25%	1,25%	1,25%
Prob. Distr. Geom. Mutação	$\frac{1}{95}$%	$\frac{1}{95}$%	$\frac{1}{95}$%	$\frac{1}{95}$%
Prob. Distr. Geom. Recombinação	$\frac{1}{95}$%	$\frac{1}{95}$%	$\frac{1}{95}$%	$\frac{1}{95}$%
Prob. Distr. Geom. Macromutação	$\frac{1}{95}$%	$\frac{1}{95}$%	$\frac{1}{95}$%	$\frac{1}{95}$%

- a elite assumiu o valor dois, mantendo-se apenas os dois melhores indivíduos de cada geração (o que se traduz em muitas das vezes em apenas um, quando os indivíduos são idênticos), de forma a garantir-se que não se daria um passo atrás em termos de qualidade;

- após experiências iniciais com outros valores, optou-se pelo valor dois para a penalização atribuída a cada aminoácido sobreposto, que acaba, aliás, por ser o valor mais comum a outros trabalhos;

- o patamar de reparação ficou-se pelos $\frac{5}{8}$ da qualidade do melhor indivíduo com o intuito de deixar, mesmo assim, indivíduos inválidos na população, contribuindo para a diversidade e, por vezes, perto de uma melhor solução, mas não deixar que indivíduos inválidos possam ser uma grande parte da população, o que poderia acontecer com as sequências maiores;

- o patamar de gerações sem evolução recebeu o valor de 50% do número mínimo

de gerações, por se considerar que era um número suficiente para que um indivíduo melhor pudesse surgir;

- as probabilidades de mutação mantiveram os valores mais usuais neste tipo de problemas, entre 1 e 2%;
- as probabilidades de recombinação mantiveram também os valores mais comuns, entre 50 e 75%;
- as probabilidades das distribuições geométricas assumiram valores baixos de maneira a garantir curvas não muito acentuadas, procurando-se não destacar muito os elementos no topo das listas (de progenitores ou do conjunto da população e da sua descendência), ao mesmo tempo que se dá possibilidade aos elementos do fim das listas de virem a ser seleccionados.

Merecem especial atenção as taxas dinâmicas dos operadores de variação. Isto deve-se essencialmente às propriedades características dos operadores de recombinação e dos operadores de mutação (estando aqui também incluídos os operadores de macromutação): a recombinação é *exploradora* do espaço compreendido entre duas soluções (os progenitores), combinando as características de ambas e actuando mais a um nível global, ao passo que a mutação é *exploratória*,[2] explorando o espaço mais próximo de uma solução, actuando mais a um nível local. Com isto em mente, é notório que, à medida que o espaço compreendido entre dois progenitores diminui, a recombinação passa também a ser menos eficiente, ao passo que a mutação continua, nessa situação, a produzir resultados; porém, enquanto os progenitores são bastante distantes (distintos), inverte-se a importância dos dois operadores [Spe93].

Portanto, quando a população tem uma maior diversidade, o operador que mais mais contribui para a criação de novos indivíduos distintos é a recombinação; já quando a população começa a estagnar e todos os indivíduos passam a ser semelhantes, a recombinação deixa de ter tanta utilidade, e é fundamentalmente através da mutação que se obtém novos indivíduos, distintos dos seus progenitores. Assim, as taxas dinâmicas funcionam como uma garantia da diversidade da população.

Como houve, *a priori*, a opção de cada um dos operadores de macromutação ter a mesma probabilidade de ser seleccionado — uma vez que só assim seria possível determinar quais os mais eficazes e promissores —, e havendo a convicção de que, para

[2]Sendo os termos em inglês mais expressivos, pretende-se associar o termo "explorador" a *"explorative"*, e o termo "exploratório" a *"exploitative"*.

algumas sequências de teste padrão, alguns operadores de macromutação deveriam ter maior taxa de utilização/aplicabilidade, optou-se pelo aumento geral da taxa de macromutação, ao longo dos vários conjuntos de teste, de forma a dar maiores oportunidades aos operadores mais úteis *nessas* sequências.

Caracterização do Equipamento

As execuções das sequências de teste padrão foram realizadas num *cluster* pertencente ao *Evolutionary and Complex Systems Group* (ECOS), cujas especificações mais notórias, à data da realização dos testes, eram:

- 1 nó servidor *Athlon 64 X2 3800+*
- 10 nós computacionais *Athlon 64 X2 3800+*
- Sistema Operativo *Ubuntu 6.06 LTS x86-64*, instalado no nó servidor
- Sistema Operativo *Ubuntu 7.04 x86-64*, instalado nos nós computacionais
- *Sun Java JDK 1.6.0-b105* (Compilador e Intérprete da Linguagem de Programação Utilizada)
- *Sun Grid Engine 6.0 DRM* (Gestor Distribuído de Recursos)

O principal factor que levou à utilização do *cluster* foi a celeridade com que foram realizados os testes. A título de exemplo, uma execução da sequência de teste padrão um, com o primeiro conjunto de parâmetros, demorava cerca de dois minutos a realizar, ao passo que num vulgar computador de secretária demorava cerca de dez minutos. Pense-se agora nos tempos de execução das sequências mais longas com o quarto conjunto de parâmetros (onde o espaço de procura era superior): no *cluster* a sequência de teste padrão doze demorava cerca de vinte-e-quatro horas a executar, e nem foi tentado realizá-la num computador de secretária. Note-se, contudo, que os programas não eram executados em paralelo no *cluster*, mas, sim, atribuídos de forma independente a cada um dos nós, se bem que a gestão fosse centralizada. No entanto, cada um dos nós estava dedicada a essa tarefa — o que não acontece num computador de utilização vulgar.

5.2 Melhores Resultados Encontrados

Alguns dos melhores resultados das execuções sobre as sequências de teste padrão, apresentadas na Tab. 3.2, com os conjuntos de parâmetros definidos na Tab. 5.1, podem ser observados nas Figs. 5.1 a 5.15, onde estão os fenótipos de alguns dos melhores indivíduos encontrados. Também alguns dos genótipos dos melhores indivíduos podem ser observados na Tab. 5.2.[3] Refira-se que cada um dos genótipos começa com a direcção *F*, frente (ou cima, conforme a interpretação), justificando-se tal facto por se ter definido que a primeira direcção seria sempre a mesma, evitando-se existirem genótipos idênticos, diferindo apenas por rotação do seu fenótipo relativamente ao ponto de origem.

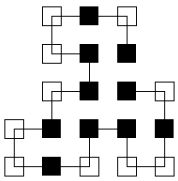


Figura 5.1: Solução encontrada para a sequência de teste padrão n.º 1

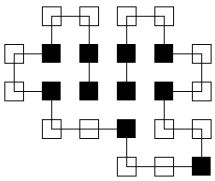


Figura 5.2: Solução encontrada para a sequência de teste padrão n.º 2

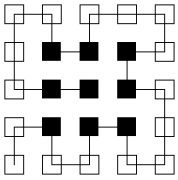


Figura 5.3: Solução encontrada para a sequência de teste padrão n.º 3

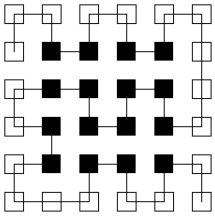


Figura 5.4: Solução encontrada para a sequência de teste padrão n.º 4

Pode-se observar que as conformações são, na sua generalidade, bastante compactas, especialmente nas situações em que os melhores resultados conhecidos foram alcançados. Mesmo as conformações para as sequências de teste padrão 7 e 8 (Figs. 5.7 e 5.8, respectivamente), apesar de não terem alcançado os melhores resultados conhecidos, mostram-se bastante compactas. Já no caso das conformações correspondentes às sequências mais longas, a 10, a 11 e a 12 (Figs. 5.10, 5.11 e 5.12, respectivamente), isso não acontece.

Na Tab. 5.2 vê-se as representações dos genótipos (em cadeias de caracteres) de

[3]Por limitações de espaço, na Tab. 5.2, as sequências com os genótipos dos indivíduos tiveram de ser colocadas em minúsculas.

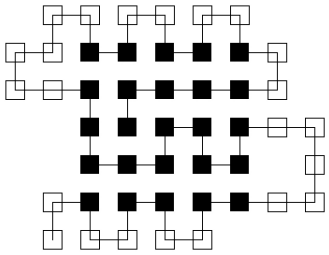


Figura 5.5: Solução encontrada para a sequência de teste padrão n.º 5

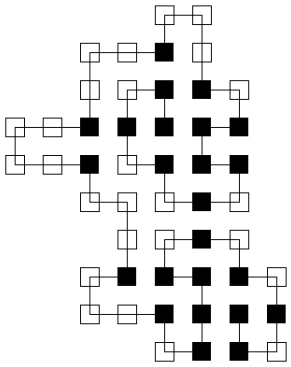


Figura 5.6: Solução encontrada para a sequência de teste padrão n.º 6

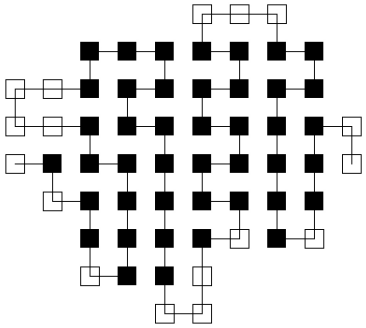


Figura 5.7: Solução encontrada para a sequência de teste padrão n.º 7

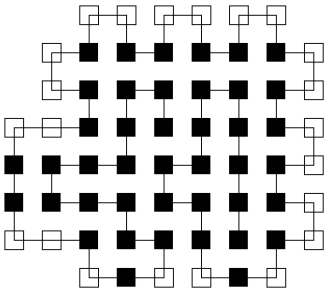


Figura 5.8: Solução encontrada para a sequência de teste padrão n.º 8

alguns dos melhores indivíduos encontrados. Na mesma tabela também podem ser observadas as energias mínimas de conformação dos melhores indivíduos encontrados e a taxa de erro (em função da diferença em pontos de energia) relativamente às melhores soluções conhecidas.

Relativamente às sequências de teste padrão, após a realização de todas as execuções com os diversos conjuntos de parâmetros, dez de quinze resultados foram igualados, ao passo que cinco deles não. Note-se, no entanto, que os resultados não igualados ficaram com uma taxa de erro inferior ou igual a 10% (a distância em pontos de energia da melhor solução encontrada à melhor solução conhecida), tal como pode ser comprovado na Tab. 5.9. Este pode ser considerado um resultado bastante aceitável, se for tido em conta que existem abordagens que se satisfazem com resultados dentro de uma margem de erro de $\frac{3}{8}$ sobre melhor resultado conhecido. Um trabalho que adopta esta margem de erro é o apresentado por Hart e Istrail em [HI95], se bem que os resultados dessa abordagem surjam de um compromisso entre a rapidez na obtenção dos resultados e a qualidade dos mesmos.

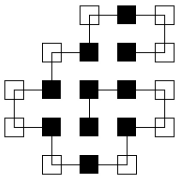


Figura 5.9: Solução encontrada para a sequência de teste padrão n.º 9

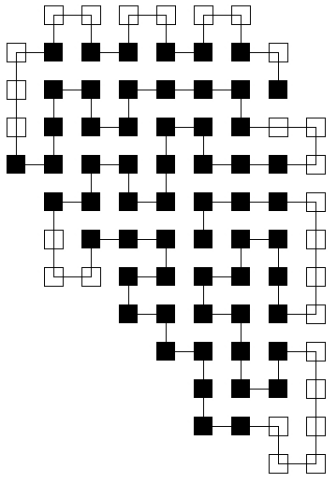


Figura 5.10: Solução encontrada para a sequência de teste padrão n.º 10

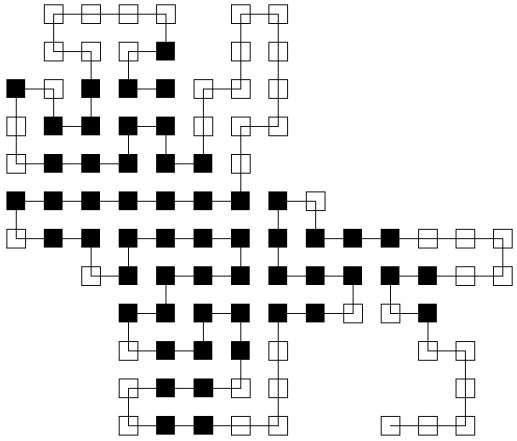


Figura 5.11: Solução encontrada para a sequência de teste padrão n.º 11

5.3 Resumo das Execuções das Sequências de Teste Padrão

De seguida faz-se um resumo dos resultados das execuções com os vários conjuntos de parâmetros para as diversas sequências de teste padrão. Nas Tabs. 5.3 a 5.8 são identificadas as sequências em análise, o melhor resultado conhecido e os melhores resultados obtidos. Para além disso, são também apresentados a média e o desvio-padrão dos melhores resultados das execuções para cada uma das sequências, bem como o número médio de gerações para cada execução, também para cada sequência.

Comece-se pela análise das Tabs. 5.3, 5.4 e 5.5, onde são apresentadas as execuções com aplicação do mecanismo de reparação e aceitando-se indivíduos inválidos (a opção

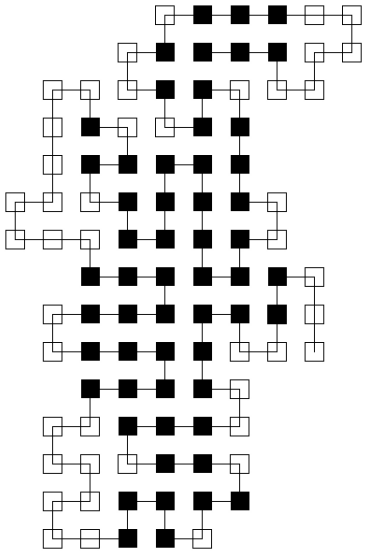


Figura 5.12: Solução encontrada para a sequência de teste padrão n.º 12

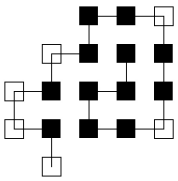


Figura 5.13: Solução encontrada para a sequência de teste padrão n.º 13

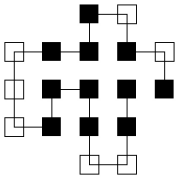


Figura 5.14: Solução encontrada para a sequência de teste padrão n.º 14

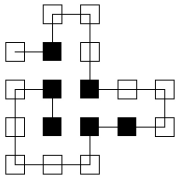


Figura 5.15: Solução encontrada para a sequência de teste padrão n.º 15

defendida), as execuções sem aplicação do mecanismo de reparação e sem aceitação de indivíduos inválidos e as execuções sem aplicar o mecanismo de reparação, mas aceitando-se indivíduos inválidos, respectivamente.

É possível notar que o número médio de gerações é bastante semelhante nas três abordagens, já que é o elemento mais independente dos vários apresentados na tabela. Passemos então às diferenças. A abordagem com melhores resultados é a com reparação e com indivíduos inválidos, onde são igualados nove resultados dos melhores conhecidos. Em segundo lugar, encontra-se a abordagem sem reparação e com indivíduos inválidos, conseguido igualar sete dos melhores resultados. Em último lugar, está a abordagem sem reparação e sem indivíduos inválidos, onde apenas seis resultados são igualados. Este ordenamento em termos de melhores resultados obtidos também se mantém para os valores não igualados, excepto para a sequência 8, onde a abordagem sem reparação e sem indivíduos inválidos é ligeiramente superior à abordagem sem reparação e com indivíduos inválidos.

Tabela 5.2: Genótipos de alguns dos melhores indivíduos encontrados

Seq.	**Genótipo**	E_{min}	**Erro**
1	*flflrrlrllflrrllfl*	**−9**	0%
2	*ffrrlrrllrrfrlfrlrrlrrf*	**−9**	0%
3	*frrllrrllflrrllflrrllflf*	**−8**	0%
4	*frrllrrllrrffrrllrrlflrrllflrrllrr*	**−14**	0%
5	*frrllrrllrffllflflrrllrfrflfrrlrrllrrllrrlrrffffl*	**−23**	0%
6	*fllflrlfllrrllrfllrflfrfllfrlrfrllfrllflrrfrlrfrr*	**−21**	0%
7	*fllffrrfffrllrlfllrrllrrllrrlfrrffffflrrllflrfllflfrlrffrrflrl*	−35	2,78%
8	*fllfrfllfrlrrfrlfrffrfllrrlrrllrrllrrlrrllrrllrrlrrfrlrrlfrrfff*	−40	4,76%
9	*frfrrlrfrlrrrlrlrfrr*	**−10**	0%
10	*frffrffrrfllrllrfllrrffrrlfrflrlrrllflrrfrlrrllfrrlfflfrlfflrrllfrrffrlrrllrrllrrlr*	−49	7,55%
11	*fflflrlrrffflfffffrllflfrrflffrffffrrffflflrfrrlrfllfflrrlfrrfffffflfrlfflfrlfrrllrffrfrrllflrrffrrll*	−45	6,25%
12	*fflfrrllflrrffflffrrlrrllrfrrllrrlrfllffrrfflfrlfrrlffrrlrrllrllfrrfflrllrfflrrlrrlrffffrrlrrlf*	−45	10,00%
13	*flrrlrlrfrffrfrrl*	**−9**	0%
14	*flrllrflfllrrfllf*	**−8**	0%
15	*fllflflrfllfrfllr*	**−4**	0%

Já olhando para as médias e para os desvios-padrão, apesar de a abordagem com reparação e com indivíduos inválidos continuar a ser a melhor (com médias de energia mais baixas), as outras duas abordagens por vezes trocam de posição. No caso das médias, nas sequências de teste padrão mais longas (especificamente a 6, a 8, a 10, a 11 e 12), a abordagem sem reparação e sem indivíduos inválidos apresenta resultados ligeiramente superiores à abordagem sem reparação e com indivíduos inválidos. No caso dos desvios-padrão, esta mudança de posições é já mais acentuada, apresentando a abordagem sem reparação e sem indivíduos inválidos resultados melhores que a outra abordagem nas sequências de tamanho maior, o que não acontece em algumas das sequências mais pequenas (nomeadamente a 1, a 3, a 4, a 10, a 13, a 14 e a 15), onde os desvios-padrão são menores. Ainda sobre os desvios-padrão, a abordagem sem reparação e sem indivíduos inválidos apresenta também valores ligeiramente menores que a abordagem com reparação e com indivíduos inválidos nas sequências 4, 5, 11 e 12.

Uma possível conclusão que se pode extrair destes factos é que as abordagens com indivíduos inválidos apresentam, ao nível das menores energias de conformação encontradas, resultados melhores, mas menos homogéneos, o que se reflecte nos desvios-padrão maiores, o que não acontece na abordagem sem indivíduos inválidos. Pode-se

Tabela 5.3: Resumo das execuções sobre as sequências de teste padrão, com o primeiro conjunto de parâmetros

N.º	E_{min}	Energia Mínima Obtida			N.º Médio de Gerações
		Melhor	Média	Desvio-Padrão	
1	**−9**	**−9**	−8,77	0,50	1009,90
2	**−9**	**−9**	−8,90	0,31	1002,00
3	**−8**	**−8**	−7,53	0,51	1014,80
4	**−14**	**−14**	−12,27	0,69	1008,23
5	**−23**	−22	−19,90	1,18	1033,87
6	**−21**	**−21**	−18,53	1,59	1023,50
7	**−36**	−34	−31,30	1,58	1016,57
8	**−42**	−36	−32,10	2,06	1067,17
9	**−10**	**−10**	−9,13	0,78	1011,23
10	**−53**	−46	−42,13	2,56	1075,77
11	**−48**	−41	−36,90	2,25	1109,30
12	**−50**	−41	−37,93	2,72	1098,60
13	**−9**	**−9**	−8,50	0,51	1002,67
14	**−8**	**−8**	−7,40	0,50	1004,33
15	**−4**	**−4**	−3,80	0,41	1002,00

inferir, então, que de facto os indivíduos inválidos injectam maior diversidade na população, levando à exploração de um espaço maior de conformações.

Por outro lado, e pegando-se agora apenas nas abordagens com indivíduos inválidos, é possível observar que a utilização do mecanismo de reparação uniformiza um pouco mais os indivíduos que a abordagem sem reparação — já que só são aceites indivíduos inválidos abaixo de um determinado patamar obtido em função do melhor indivíduo. Resulta, então, que o mecanismo de reparação, nas condições em que é utilizado (onde indivíduos inválidos são parcialmente aceites), permite uma maior diversidade da população, com uma maior exploração do espaço de conformações, mas não deixa que a aceitação de indivíduos inválidos não parta para uma zona do espaço onde abundem as conformações inválidas, enviesando, de forma nefasta, os resultados obtidos. Ou seja, o espaço de procura centra-se nos indivíduos válidos e naqueles que, mesmo inválidos, se encontram bastante próximos da "fronteira" com os indivíduos válidos.

Em complemento aos dados apresentados nas tabelas com o resumo das execuções, podem ser observados em anexo, constante no CD, os gráficos com a representação dos melhores indivíduos encontrados com cada conjunto de parâmetros, para as várias

Tabela 5.4: Resumo das execuções sobre as sequências de teste padrão, com o primeiro conjunto de parâmetros, sem aplicação do mecanismo de reparação e sem indivíduos inválidos

N.º	E_{min}	**Energia Mínima Obtida**			**N.º Médio de Gerações**
		Melhor	**Média**	**Desvio-Padrão**	
1	**−9**	**−9**	−7,53	0,90	1009,87
2	**−9**	**−9**	−7,97	0,67	1007,87
3	**−8**	**−8**	−6,27	0,83	1002,00
4	**−14**	−13	−11,33	1,12	1025,00
5	**−23**	−20	−16,70	1,53	1035,27
6	**−21**	−18	−15,30	1,58	1014,53
7	**−36**	−31	−28,57	1,87	1084,90
8	**−42**	−33	−27,60	2,25	1055,03
9	**−10**	**−10**	−8,20	0,89	1007,40
10	**−53**	−43	−38,27	2,56	1029,13
11	**−48**	−38	−34,93	2,13	1060,03
12	**−50**	−40	−34,43	2,64	1076,60
13	**−9**	−8	−7,63	0,49	1002,00
14	**−8**	**−8**	−7,07	0,52	1007,80
15	**−4**	**−4**	−3,17	0,59	1004,73

sequências de teste padrão.

5.4 Comparação de Resultados

De seguida, pode ser observada na Tab. 5.9 a comparação dos melhores resultados obtidos com a solução proposta, a GARMM *(Genetic Algorithm with a Repair Mechanism and Macromutations)*, com os melhores resultados de outras abordagens, especificamente a PFGA *(Protein Folding Genetic Algorithm)*, a ACO *(Ant Colony Optimization)*, a EMC *(Evolutionary Monte Carlo)*, a GTS *(Genetic algorithm combined with Tabu Search)*, a MMA *(Multimeme Algorithm)*, a MMC *(Metropolis Monte Carlo)* e a GA *(Genetic Algorithm)*.

Observando-se a Tab. 5.9, pode-se notar que a abordagem proposta, apesar não ser a melhor delas — fica-se pelo terceiro lugar —, é bastante competitiva com as actuais abordagens. Apresenta resultados bastante bons para uma abordagem relativamente simples, em termos do algoritmo genético em si, quando comparada com a ACO [SH03] ou com a PFGA [BS05], que utilizam estruturas secundárias e optimizações locais, respectivamente, para auxiliarem na obtenção da conformação de

Tabela 5.5: Resumo das execuções sobre as sequências de teste padrão, com o primeiro conjunto de parâmetros, sem aplicação do mecanismo de reparação, mas com indivíduos inválidos

N.º	E_{min}	Energia Mínima Obtida			N.º Médio de Gerações
		Melhor	Média	Desvio-Padrão	
1	**−9**	**−9**	−8,00	0,83	1004,83
2	**−9**	**−9**	−8,30	0,70	1002,00
3	**−8**	**−8**	−6,67	0,71	1004,03
4	**−14**	−13	−11,47	1,04	1009,07
5	**−23**	−21	−17,57	1,55	1024,10
6	**−21**	−19	−14,50	3,60	1018,77
7	**−36**	−33	−28,57	2,67	1056,87
8	**−42**	−32	−27,40	3,50	1055,03
9	**−10**	**−10**	−8,80	0,89	1005,93
10	**−53**	−45	−37,93	3,88	1100,80
11	**−48**	−39	−31,10	5,38	1080,17
12	**−50**	−40	−34,07	3,89	1087,73
13	**−9**	**−9**	−8,00	0,37	1002,00
14	**−8**	**−8**	−7,13	0,43	1002,00
15	**−4**	**−4**	−3,53	0,51	1009,23

proteínas.

Comparativamente às restantes abordagens, com um nível de complexidade da ordem daquele da abordagem proposta, os resultados revelam-se iguais ou superiores, provando que a utilização de um mecanismo cujo único intuito é reparar indivíduos inválidos revela resultados promissores.

5.5 Operadores de Macromutação

À partida, a razão para a utilização dos operadores de macromutação seria a crença (ver, por exemplo, [KHSP99]) de que há uma série de padrões nas sequências de direcções (nos indivíduos) que se revelam comuns em muitos deles, e mesmo em sequências de aminoácidos distintas, que mais vale tratar como um todo, fazendo-se mutações localizadas e específicas, em vez de puramente aleatórias. Outra justificação seria também crer-se que haveria alguma vantagem em manipular conjuntos de genes em simultâneo (mesmo sem a verificação de padrões), ao invés de apenas mutar um gene (ou poucos mais) de forma isolada, como muitas vezes acontece com o operador

Tabela 5.6: Resumo das execuções sobre as sequências de teste padrão, com o segundo conjunto de parâmetros

N.º	E_{min}	**Energia Mínima Obtida**			**N.º Médio de Gerações**
		Melhor	**Média**	**Desvio Padrão**	
5	**−23**	**−23**	−20,00	1,31	2051,23
7	**−36**	−34	−31,93	1,14	2094,10
8	**−42**	−40	−33,57	2,56	2054,93
10	**−53**	−48	−44,13	2,16	2119,57
11	**−48**	−43	−39,80	2,66	2095,93
12	**−50**	−44	−40,17	1,86	2091,33

Tabela 5.7: Resumo das execuções sobre as sequências de teste padrão, com o terceiro conjunto de parâmetros

N.º	E_{min}	**Energia Mínima Obtida**			**N.º Médio de Gerações**
		Melhor	**Média**	**Desvio Padrão**	
7	**−36**	−35	−32,40	1,30	3037,40
8	**−42**	−38	−34,33	2,06	3072,13
10	**−53**	−49	−44,83	2,74	3091,93
11	**−48**	−44	−40,17	2,02	3094,37
12	**−50**	−45	−41,20	1,95	3129,43

de mutação mais usual.

O primeiro aspecto é confirmado, por exemplo, pela observação da Tab. 5.10, em particular no caso dos operadores *manivela*, *dobragem*, *vincagem*, *rotação*, *serpenteação* e *desdobragem*, que são aplicados se e só se determinados padrões (a serem identificados por cada um dos macromutadores) forem encontrados no genótipo do indivíduo. Dependendo das sequências de teste padrão utilizadas, cerca de 2% a 6% dos indivíduos da descendência — que neste caso correspondia a 675 indivíduos — de cada geração apresentava um dos padrões *procurados* por estes macromutadores. Se for tido em conta que a probabilidade de aplicação destes operadores de macromutação variava entre 10% e 20% e que os operadores de macromutação com busca de padrões correspondiam a $\frac{6}{11}$ da probabilidade de aplicar um operador de macromutação, é-se levado a crer que o número de indivíduos que apresentava padrões susceptíveis de serem macromutados possa ser, pelo menos, nove vezes superior.

Tabela 5.8: Resumo das execuções sobre as sequências de teste padrão, com o quarto conjunto de parâmetros

N.º	E_{min}	Energia Mínima Obtida			N.º Médio de Gerações
		Melhor	Média	Desvio Padrão	
7	**−36**	−34	−32,27	1,26	4108,80
8	**−42**	−39	−34,43	1,85	4010,10
10	**−53**	−49	−45,63	1,97	4123,10
11	**−48**	−45	−40,73	1,86	4089,10
12	**−50**	−45	−41,17	1,93	4084,03

Outro resultado de certa forma surpreendente é o facto de os operadores de macromutação se revelarem menos destrutivos que o simples operador de mutação. Tal pode ser constatado na Tab. 5.11 (onde não houve lugar a reparação, nem foram aceites indivíduos inválidos) onde os operadores de macromutação, apesar de por vezes terem uma expressão baixa, se revelam muito mais utilizados que a mutação simples, cujos resultados se revelam próximos de zero, situando-se na casa das milésimas. Faça-se, no entanto, a ressalva de que o operador de mutação só seria aplicado se os operadores de macromutação tivessem sido aplicados sem sucesso, o que pode justificar, em parte, os valores próximos de zero.

Algo que se pode inferir destes resultados é que as sequências rapidamente evoluem para enrolamentos de tal forma complexos em que alterações localizadas, como é o caso dos operadores de macromutação, causam menores alterações de âmbito global ao fenótipo de um indivíduo que o operador de mutação simples, que ao mudar um simples gene (por exemplo a meio da sequência) poderá estar a sobrepor parte da cadeia dos aminoácidos a outra parte. Isto, de certa forma, acaba por estar também de acordo com o pressuposto inicial.

Outro aspecto que interessa focar é a razão de haver onze operadores de macromutação, para além do operador de mutação simples. Dos operadores de macromutação utilizados, alguns eram já conhecidos [RCMJJ04] (*rotação*, *serpenteação*, *manivela* e *vincagem*), todos eles com busca e substituição de padrões. Mas então colocava-se a seguinte questão: será que não haveria outros macromutadores que pudessem também ser úteis? Foi com este intuito que foram adicionados dois novos macromutadores com busca de padrões (*dobragem* e *desdobragem*, bem como mais cinco macromutadores sem busca de padrões, mas com substituição localizada (*inserção*, *inversão*, *baralhação*, *troca* e *troca-sequência*).

Tabela 5.9: Comparação dos melhores resultados obtidos por diferentes abordagens às sequências de teste padrão estudadas

Seq.	E_{min}	**GARMM**	**PFGA**	**ACO**	**EMC**	**GTS**	**MMA**	**MMC**	**GA**
1	**−9**	**−9**	**−9**	**−9**	**−9**	**−9**	**−9**	**−9**	**−9**
2	**−9**	**−9**	**−9**	**−9**	**−9**	**−9**	–	**−9**	**−9**
3	**−8**	**−8**	**−8**	**−8**	**−8**	**−8**	**−8**	**−8**	**−8**
4	**−14**	**−14**	**−14**	**−14**	**−14**	**−14**	**−14**	−13	−12
5	**−23**	**−23**	**−23**	**−23**	**−23**	**−23**	−22	−20	−22
6	**−21**	**−21**	**−21**	**−21**	**−21**	**−21**	**−21**	**−21**	**−21**
7	**−36**	−35	**−36**	**−36**	−35	−35	**−36**	−33	−34
8	**−42**	−40	**−42**	**−42**	−39	−39	−38	−35	−37
9	**−10**	**−10**	–	–	–	–	–	–	–
10	**−53**	−49	**−53**	−51	–	–	–	–	–
11	**−48**	−45	**−48**	−47	–	–	–	–	–
12	**−50**	−45	−49	−47	–	–	–	–	–
13	**−9**	**−9**	–	–	–	–	–	–	–
14	**−8**	**−8**	–	–	–	–	–	–	–
15	**−4**	**−4**	–	–	–	–	–	–	–

De maneira a não enviesar a análise de cada um dos macromutadores, todos eles tiveram a mesma probabilidade de ser aplicados. E disso surge que, dos macromutadores com busca de padrões, o mais utilizado é o *desdobragem*. Não é de todo estranho que tal aconteça, já que o que este faz é "alisar" parte da sequência (evitando-se assim algumas sobreposições de aminoácidos). Fica ainda por analisar — o que poderá ser feito em trabalho futuro — se o seu efeito não será contraproducente. Depois, os operadores *serpenteação*, *manivela* e *vincagem* apresentam resultados de aplicação semelhantes, ao passo que o operador *rotação* revela ser o menos aplicado, o que também não surpreende, já que rodar parte de uma conformação, por exemplo, da esquerda para a direita, é bastante provável que dê origem a sobreposições de aminoácidos.

Quanto ao operador *dobragem*, este é talvez aquele com um comportamento mais oscilatório, com uma aplicação a variar entre as centésimas e as unidades. É compreensível que a sua taxa de aplicação não seja das maiores, já que o que faz é pegar num excerto "liso" e fazer-lhe uma dobra, surgindo um "alto", o que pode levar a sobreposições dos aminoácidos que ficaram nesse "alto" com os existentes na vizinhança. Só assim se poderá justificar o aumento da sua taxa de aplicação nas sequências maiores, em que o espaço de conformação é maior e é menos provável a criação de novas sobreposições (pelo menos enquanto a conformação não atingir a sua forma

Tabela 5.10: Utilização média dos vários operadores de variação por geração, com primeiro conjunto de parâmetros, nas várias sequências de teste padrão, para 30 execuções

Operadores	**Sequências**														
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Recombinação	194,20	195,49	200,25	200,39	207,43	207,72	218,69	223,17	193,96	224,76	228,39	227,64	194,27	192,94	193,36
Manivela	9,32	9,99	5,04	8,47	9,11	8,43	7,66	7,84	6,81	7,96	7,73	7,84	9,00	5,37	1,13
Dobragem	0,06	0,20	3,36	2,09	3,23	2,72	3,13	3,09	0,03	4,21	4,58	4,04	0,03	0,19	0,14
Vincagem	6,06	9,43	5,47	8,25	8,83	8,89	8,16	8,23	9,54	8,13	7,92	7,98	8,00	8,30	9,43
Rotação	0,05	0,37	0,10	0,37	1,07	1,56	0,90	2,42	3,04	1,96	3,08	3,29	1,99	0,32	0,30
Serpenteação	5,06	6,75	3,98	6,24	7,75	6,31	6,70	6,89	0,58	7,02	6,95	7,07	1,06	1,77	0,71
Desdobragem	10,13	10,24	8,14	9,55	9,44	9,28	8,60	8,28	8,69	8,21	7,92	8,00	9,47	7,63	1,90
Σ *parcial*	30,68	36,98	26,09	34,97	39,43	37,19	35,15	36,75	28,69	37,49	38,18	38,22	29,55	23,58	13,61
Inserção	8,37	8,59	8,37	8,75	8,56	8,52	7,97	7,66	8,43	7,73	7,52	7,55	8,23	8,30	8,28
Inversão	8,40	8,59	8,35	8,74	8,56	8,56	7,93	7,68	8,40	7,70	7,55	7,57	8,21	8,31	8,27
Baralhação	5,59	6,05	5,89	6,87	7,09	7,18	6,80	6,62	5,70	6,86	6,81	6,84	5,15	5,26	5,39
Troca	9,84	9,86	9,56	9,68	9,27	9,22	8,49	8,19	9,85	8,11	7,85	7,90	9,77	9,85	9,82
Troca-Sequência	10,42	10,34	9,98	9,95	9,45	9,42	8,66	8,32	10,41	8,22	7,93	8,01	10,39	10,55	10,49
Σ *parcial*	42,62	43,43	42,15	43,99	42,93	42,9	39,85	38,47	42,79	38,62	37,66	37,87	41,75	42,27	42,25
Σ *macromutações*	73,30	80,41	68,24	78,96	82,36	80,09	75,00	75,22	71,48	76,11	75,84	76,09	71,30	65,85	55,86
Mutação	157,48	182,32	187,52	248,82	294,16	303,10	326,06	332,04	158,21	389,87	418,98	419,78	142,77	145,43	147,60
Reparação	336,06	396,34	383,85	329,05	442,52	399,31	369,94	484,09	223,96	489,80	486,44	486,93	325,12	346,88	255,06

Tabela 5.11: Utilização média dos vários operadores de variação por geração, com primeiro conjunto de parâmetros, nas várias sequências de teste padrão, sem utilização do mecanismo de reparação e sem indivíduos inválidos

Operadores	**Sequências**														
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Recombinação	135,84	101,20	116,32	99,62	94,87	68,43	121,62	99,74	151,67	95,55	41,77	43,18	116,49	154,39	119,67
Manivela	3,59	1,25	2,11	2,67	2,90	2,67	1,76	3,34	0,20	2,29	2,62	3,10	1,17	2,20	1,13
Dobragem	0,05	0,13	0,18	0,27	0,13	0,29	0,59	0,64	0,07	0,32	0,39	0,77	0,05	0,03	0,17
Vincagem	4,32	5,35	5,04	4,56	5,74	4,72	3,35	5,57	1,32	4,00	4,63	4,15	2,96	2,96	6,35
Rotação	0,06	0,26	0,07	0,34	0,64	0,23	0,10	0,31	0,13	0,51	0,35	0,45	0,05	0,30	0,06
Serpenteação	3,26	4,87	3,28	5,65	4,78	3,29	3,01	4,46	0,70	4,97	4,87	5,80	1,60	1,92	2,45
Desdobragem	8,45	8,17	8,93	8,46	6,20	6,59	6,28	5,65	4,72	5,92	6,39	5,69	9,08	7,73	4,96
Σ *parcial*	19,73	20,03	19,61	21,95	20,39	17,79	15,09	19,97	7,14	18,01	19,25	19,96	14,91	15,14	15,12
Inserção	3,10	3,23	4,02	4,03	2,29	2,20	2,45	2,04	2,52	1,93	2,01	1,96	3,67	3,55	3,59
Inversão	3,84	3,63	3,91	3,32	2,39	2,49	2,15	2,03	3,86	1,75	1,76	1,75	4,04	3,93	4,45
Baralhação	1,69	1,37	2,40	2,26	1,21	1,16	1,23	0,92	1,52	0,88	0,81	0,78	1,89	1,90	1,75
Troca	5,05	4,97	5,65	5,24	4,14	4,27	3,90	3,74	5,41	3,54	3,62	3,59	5,78	5,63	5,47
Troca-Sequência	3,55	3,04	3,58	2,70	1,53	1,43	1,19	1,02	3,87	0,73	0,74	0,76	4,47	4,21	3,88
Σ *parcial*	17,23	16,24	19,56	17,55	11,56	11,55	10,92	9,75	17,18	8,83	8,94	8,84	19,85	19,22	19,14
Σ *macromutações*	36,96	36,27	39,17	39,50	31,95	29,34	26,01	29,72	24,32	26,84	28,19	28,8	34,76	34,36	34,26
Mutação	0,002	0,003	0,002	0,002	0,003	0,003	0,004	0,004	0,002	0,004	0,004	0,004	0,002	0,002	0,003

Tabela 5.12: Utilização média dos vários operadores de variação por geração, com o primeiro conjunto de parâmetros, nas várias sequências de teste padrão, sem utilização do mecanismo de reparação, mas com indivíduos inválidos

Operadores	**Sequências**														
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Recombinação	197,94	198,97	198,23	203,08	210,77	205,76	218,66	223,61	196,59	229,04	226,44	229,61	194,66	192,26	193,81
Manivela	7,84	8,72	4,44	7,86	8,38	7,75	7,53	7,95	5,76	7,71	7,88	7,67	8,99	4,12	1,03
Dobragem	0,07	0,50	2,40	2,82	3,17	2,32	3,12	2,84	0,10	4,05	4,23	3,79	0,03	0,14	0,57
Vincagem	7,31	8,42	5,43	7,82	8,81	9,44	8,33	8,16	9,42	7,85	8,08	7,85	6,66	7,90	8,48
Rotação	0,16	0,56	0,33	0,53	1,20	2,36	1,14	1,55	2,94	1,78	2,78	3,09	0,05	0,40	0,33
Serpenteação	4,40	5,21	4,64	5,92	7,29	4,92	6,40	6,41	0,76	6,86	6,82	6,74	1,24	3,01	1,14
Desdobragem	9,45	9,61	8,18	9,36	9,14	9,08	8,38	8,29	8,83	7,88	8,08	7,83	9,70	7,58	2,57
Σ *parcial*	29,23	33,02	25,42	34,31	37,99	35,87	34,90	35,20	27,81	36,13	37,87	36,97	26,67	23,15	14,12
Inserção	8,16	8,39	8,47	8,58	8,31	8,68	7,94	7,69	8,26	7,41	7,63	7,40	8,19	8,30	8,26
Inversão	8,20	8,37	8,48	8,58	8,34	8,64	7,95	7,67	8,26	7,40	7,64	7,45	8,19	8,35	8,25
Baralhação	5,47	5,89	6,03	6,74	6,92	7,25	6,80	6,60	5,62	6,62	6,93	6,70	5,23	5,29	5,32
Troca	9,60	9,60	9,66	9,48	9,04	9,38	8,52	8,16	9,66	7,78	8,00	7,76	9,72	9,90	9,77
Troca-Sequência	10,13	10,04	10,09	9,79	9,18	9,55	8,66	8,30	10,24	7,88	8,06	7,83	10,40	10,56	10,45
Σ *parcial*	41,56	42,29	42,73	43,17	41,79	43,5	39,87	38,42	42,04	37,09	38,26	37,14	41,73	42,40	42,05
Σ *macromutações*	70,79	75,31	68,15	77,48	79,78	79,37	74,77	73,62	69,85	73,22	76,13	74,11	68,4	65,55	56,17
Mutação	154,35	179,98	189,79	245,62	289,92	307,27	326,63	332,13	155,86	384,77	422,06	417,80	143,18	146,10	147,04

Tabela 5.13: Utilização média dos vários operadores de variação por geração, com o segundo conjunto de parâmetros, nas várias sequências de teste padrão, para 30 execuções

Operadores	**Sequências**					
	5	**7**	**8**	**10**	**11**	**12**
Recombinação	390,55	405,73	406,31	419,76	421,69	432,96
Manivela	30,63	28,32	28,65	27,71	27,50	26,04
Dobragem	7,11	13,28	10,29	13,04	14,56	13,39
Vincagem	27,32	28,74	28,80	27,73	27,59	26,34
Rotação	3,24	2,91	3,46	4,34	7,26	7,81
Serpenteação	22,88	25,03	25,81	26,70	24,65	25,04
Desdobragem	30,93	29,29	29,21	27,80	27,62	26,39
Σ *parcial*	122,11	127,57	126,22	127,32	129,18	125,01
Inserção	28,02	26,98	27,06	26,13	26,13	24,98
Inversão	28,07	26,99	27,06	26,13	26,18	24,98
Baralhação	23,23	23,05	23,34	23,27	23,65	22,59
Troca	30,34	28,87	28,77	27,46	27,34	26,09
Troca-Sequência	31,01	29,34	29,28	27,83	27,62	26,38
Σ *parcial*	140,67	135,23	135,51	130,82	130,92	125,02
Σ *macromutações*	262,78	262,80	261,73	258,14	260,10	250,03
Mutação	571,96	635,05	659,90	750,99	807,77	797,65
Reparação	873,54	690,25	857,79	783,26	856,85	861,84

mais compacta).

Passando-se aos macromutadores sem busca de padrões, os mais promissores são o *troca* e *inversão* — os menos destrutivos —, já que apenas se trocam dois genes (no primeiro) ou se altera a ordem a dos genes num excerto bem delimitado (no segundo), sabendo-se à partida que esse excerto não foi alterado (por exemplo, se não existiam antes, também não passou a haver sobreposições nesse excerto). Estes operadores podem, no entanto, criar novas sobreposições noutros locais do fenótipo, mas tal deve ser raro, se se prestar atenção aos seus valores de aplicação. Já o operador menos utilizado é o *baralhação*, talvez o mais destrutivo em termos do genótipo dos indivíduos, alterando a ordem de uma série de genes, podendo ele mesmo criar sobreposições nesse excerto.

Em função do exposto anteriormente e da análise das Tabs. 5.10 e 5.11, é de crer que a aplicação de operadores de macromutação será menos destrutiva que o simples uso do operador de mutação, uma vez que são substituídos excertos por

Tabela 5.14: Utilização média dos vários operadores de variação por geração, com o terceiro conjunto de parâmetros, nas várias sequências de teste padrão, para 30 execuções

Operadores	**Sequências**				
	7	**8**	**10**	**11**	**12**
Recombinação	595,41	597,84	617,72	621,42	614,80
Manivela	58,50	59,75	57,12	56,77	57,17
Dobragem	27,29	20,46	31,93	25,99	31,37
Vincagem	59,39	59,97	57,34	56,86	57,82
Rotação	2,08	10,25	3,71	12,46	17,32
Serpenteação	42,10	54,77	47,60	49,93	53,48
Desdobragem	60,49	60,27	57,37	56,86	57,84
Σ *parcial*	249,85	265,47	255,07	258,87	275,00
Inserção	55,85	55,77	53,97	53,91	54,79
Inversão	55,82	55,74	53,93	53,89	54,78
Baralhação	47,71	48,16	48,03	48,67	49,42
Troca	59,63	59,31	56,76	56,29	57,27
Troca-Sequência	60,65	60,33	57,44	56,93	57,82
Σ *parcial*	279,66	279,31	270,13	269,69	274,08
Σ *macromutações*	529,51	544,78	525,20	528,56	549,08
Mutação	887,32	908,74	1041,89	1116,45	1109,22
Reparação	1070,70	1319,59	1193,59	1314,51	1296,64

outros excertos, não afectando a proteína de uma forma radical. Aliás, a observação da Tab. 5.11 (onde os operadores só eram aceites se não criassem indivíduos inválidos) pode apoiar esta afirmação, notando-se que a aplicabilidade do operador de mutação simples é bastante reduzido, em termos médios na casa das milésimas, quase não tendo expressão. É possível também observar que os operadores de macromutação com busca de padrões são mais destrutivos que os sem busca de padrões; no entanto, têm uma aplicabilidade mais específica e localizada, baseando-se em padrões comuns dentro de conformações, o que justifica a sua utilização.

5.6 Taxas Dinâmicas de Variação

No que às taxas dinâmicas de variação diz respeito, realça-se, mais uma vez, a sua real aplicabilidade. Para além do que já foi sendo dito acerca a importância da recombinação, para evolução de soluções quando a população é mais heterogénea, e da importância da mutação, para a evolução quando a população é mais homogénea,

Tabela 5.15: Utilização média dos vários operadores de variação por geração, com o quarto conjunto de parâmetros, nas várias sequências de teste padrão, para 30 execuções

Operadores	**Sequências**				
	7	**8**	**10**	**11**	**12**
Recombinação	789,92	799,94	809,51	819,97	805,10
Manivela	55,90	59,26	58,69	57,54	59,41
Dobragem	18,35	19,65	27,74	28,26	30,91
Vincagem	59,58	59,71	58,51	57,62	59,22
Rotação	3,12	9,37	11,33	13,68	18,77
Serpenteação	42,13	53,18	54,21	51,64	53,86
Desdobragem	60,97	59,87	58,93	57,80	59,41
Σ *parcial*	240,05	261,04	269,41	266,54	281,58
Inserção	56,17	55,40	55,41	54,71	56,26
Inversão	56,17	55,35	55,40	54,75	56,25
Baralhação	48,15	47,88	49,31	49,42	50,86
Troca	60,03	58,98	58,22	57,18	58,83
Troca-Sequência	61,08	60,01	58,92	57,79	59,45
Σ *parcial*	281,6	277,62	277,26	273,85	281,65
Σ *macromutações*	521,65	538,66	546,67	540,39	563,23
Mutação	1305,28	1328,31	1524,43	1629,83	1634,93
Reparação	1425,59	1833,79	1622,82	1755,24	1722,49

interessa referir a adaptação dos operadores de variação às sequências de teste padrão, só permitida pela utilização de taxas dinâmicas.

Tal adaptação pode ser comprovada observando-se qualquer uma das tabelas com os resultados das execuções, mas é talvez mais evidente na Tab. 5.10. Ignore-se o operador de mutação, já que este, mesmo que fosse fixo, mostraria valores oscilantes de aplicação de uma sequência de teste padrão para outra, uma vez que está dependente essencialmente dos tamanhos da sequência — quanto maior for a sequência, maior será, em termos absolutos, o valor da sua aplicação.

Quanto aos operadores de macromutação, este mostra valores absolutos de aplicação oscilantes que aparentam ser dependentes de dois factores: o tamanho da sequência e a própria sequência em si. Fazendo novamente referência à Tab. 5.10, é possível notar que nas sequências mais pequenas o valor de aplicação dos macromutadores com busca de padrões é inferior ao das sequências mais longas, tal como acontece com o operador de mutação convencional. No entanto, em sequências com o mesmo

tamanho, como é o caso das sequências 13, 14 e 15 (todas elas com 18 de tamanho), a aplicação dos macromutadores com busca de padrões têm valores de aplicação bastante díspares: 29,55, 23,58 e 13,61, respectivamente. Na mesma linha, a sequência que tem uma maior aplicabilidade destes macromutadores é a sequência 5, com um valor de 39,43, ligeiramente superior ao das sequências mais longas, mas com metade do tamanho destas. Tal leva a crer que, para além do número de aminoácidos de uma proteína, também a própria sequência de aminoácidos tem uma importância elevada no processo de conformação.

Também nos operadores de macromutação sem busca de padrões é possível observar que a sua real aplicação não depende apenas só do tamanho da sequência — as últimas três sequências, as mais pequenas, continuam a ter uma aplicação mais baixa — mas também da sequência em si. Em sequências com o mesmo tamanho continua a haver valores díspares de aplicação: as sequências 13, 14 e 15 têm os valores 71,30, 65,85, e 55,86, respectivamente. E, tal como acontece com os macromutadores com busca de padrões, não são necessariamente as sequências mais longas aquelas que têm um valor de aplicação superior: a sequência 5 continua a ter o maior valor de aplicação, acompanhada pelas sequências 4 e 6, que têm também sensivelmente metade do tamanho das sequências mais longas. Logo, reforça-se a ideia de que a própria sequência é importante.

Já o operador de recombinação, se tivesse uma taxa de aplicação fixa, apresentaria sensivelmente os mesmos valores totais médios de aplicação para cada uma das sequências de teste padrão, visto que os valores da sua aplicação dependem essencialmente do número de indivíduos a que pode ser aplicado — i.e., do tamanho da população —, e não do tamanho das sequências. No entanto, crê-se que devido ao facto da taxa de recombinação ser variável, não é isso o que acontece no algoritmo proposto. Se se prestar atenção à Tab. 5.10, é possível notar que o valor de aplicação do operador de recombinação, ao invés de se manter constante (já que o tamanho da população também se mantém), vai oscilando de acordo com o tamanho da sequência. Este valor varia entre aproximadamente 190, para as sequências mais pequenas, e aproximadamente 230, para as sequências mais longas. Por outro lado, tanto quanto é possível observar, a sequência dos aminoácidos em si, para além do seu tamanho, não tem qualquer peso na aplicação da recombinação.

Assim, o valor absoluto de aplicação da recombinação, em vez de se manter constante em cada uma das sequências, varia também em função do tamanho das sequências. Sequências mais longas originam execuções mais longas do algoritmo,

mais pontos de estagnação e, consequentemente, mais alterações nas taxas dinâmicas de variação. Com base nesta observação, é possível concluir que sequências mais curtas mais facilmente originam populações mais homogéneas, favorecendo-se aqui os operadores de mutação, ao passo que aumento do tamanho das sequências leva a que seja privilegiado o operador de recombinação. Ou seja, a utilização de taxas dinâmicas de variação permite ao algoritmo genético "favorecer" um operador de variação em detrimento de outros, em função das sequências de teste estudadas — ou dos seus tamanhos.

Curiosamente, os dados presentes na Tab. 5.11 (onde não existe reparação nem são aceites indivíduos inválidos) apresentam resultados bastante distintos, quase opostos, dos das Tabs. 5.10 e 5.12 (com reparação e sem reparação, respectivamente, mas onde são aceites indivíduos inválidos). Tal leva a pressupor que a aceitação ou não aceitação de indivíduos inválidos na população também influencia a evolução da população, os seus pontos de estagnação e, logo, as alterações dos valores das taxas dinâmicas de variação. O que não surpreende, visto que o espaço de procura com indivíduos inválidos é bastante superior ao espaço de procura com indivíduos válidos, onde mais facilmente a população fica homogénea (ou lá próximo) e, logo, mais frequentemente são alterados os valores das taxas de variação — relembre-se que a taxa de recombinação desce e as taxas de mutação e macromutação sobem.

5.7 Importância dos Tamanhos e dos Padrões das Sequências

Observando-se novamente a Tab. 5.3, nomeadamente as colunas com os melhores resultados, médias e desvios-padrão, a primeira conclusão que se pode tirar, e também a mais óbvia, é que o tamanho das sequências de testes padrão desempenha o papel mais importante na dificuldade de obtenção do melhor indivíduo. Se se prestar atenção aos valores para as sequências de teste padrão 1, 2, 3, 4, 6, 9, 13, 14 e 15, que são as mais pequenas, pode ser notado que os melhores resultados obtidos são mais homogéneos, com menores desvios relativos à média de valores; por outro lado, nas sequências maiores, 7, 8, 10, 11 e 12 pode-se notar que os valores para as energias mínimas são bastante mais heterogéneos. E tal é válido para qualquer dos conjuntos de parâmetros estudados: mesmo quando os resultados para as sequências maiores são melhorados, os melhores valores obtidos em cada execução continuam a ser pouco

uniformes, havendo uma grande discrepância entre o pior e o melhor resultado obtido — o que não é de estranhar, dado que quanto maior for a sequência, maior é o universo de soluções possíveis.

A título de exemplo, em [UM93] é apresentado o número total de conformações possíveis para a primeira sequência de teste padrão, sendo este 83.779.155. Destas conformações, 4 são para o melhor resultado, -9, e 36.098.079 para o pior resultado, 0, encontrando-se os restantes resultados numa ordem de valores decrescente até ao melhor resultado — sendo que não são aceites indivíduos inválidos. Ora, se se tem estes valores para uma das sequências de teste padrão mais pequenas, com 20 aminoácidos, bastante mais complexo será o problema para sequências de 85 ou 100 aminoácidos, as maiores dos testes padrão, o que ajuda a explicar as dificuldades existentes (mais ainda nesta situação, onde são também aceites indivíduos inválidos).

Uma interessante excepção é a sequência 5. Possui sensivelmente o mesmo comprimento que a sequência 6 — a sequência 5 com 48 aminoácidos e a sequência 6 com 50 aminoácidos — e uma energia mínima de conformação também próxima da da sequência 6 (-23 e -21, respectivamente), mas só com um maior espaço de procura foi possível encontrar a melhor solução (e, mesmo assim, as melhores soluções para cada execução continuam a ser bastante díspares). Isto é mais um indício de que, para além do tamanho da sequência, também importa, em grande medida, a sua estrutura interna — i.e., o alinhamento dos aminoácidos e os padrões aí existentes. Foi aliás com base em padrões do alinhamento interno que as abordagens PFGA [BS05] e ACO [SH03] capitalizaram para apresentar melhores resultados.

Ainda sobre a importância da estrutura interna de uma sequência na obtenção da sua conformação, recordem-se as observações feitas quanto aos valores da aplicação dos operadores de macromutação, na *Secção 5.6*.

Escalabilidade

Tendo em conta o exposto anteriormente, a abordagem proposta sofre das mesmas limitações de todas as outras abordagens conhecidas: com o aumento das sequências de aminoácidos em estudo e, parcialmente, com a complexidade intrínseca à estrutura, começa a apresentar resultados com uma taxa de erro diferente de zero e com tendência a aumentar.

Contudo, como este é um problema generalizado, acredita-se na validade da abordagem — que mostrou resultados satisfatórios em termos comparativos — e também

na sua evolução, eventualmente recorrendo a reparações localizadas, fazendo o paralelismo com as abordagens que trabalham localmente.

5.8 Mecanismo de Reparação

Analisemos agora os benefícios da utilização de um mecanismo de reparação na abordagem ao problema estudado. Para se poder tirar conclusões sobre o mesmo, em termos comparativos, um dos passos a tomar foi executar o algoritmo sem recurso ao mecanismo de reparação. Tendo-se *a priori* resultados de um algoritmo genético sem qualquer mecanismo de reparação — uma abordagem com um algoritmo genético clássico, apresentada em [UM93] — que permitiria tirar, à partida, algumas conclusões, optou-se por pegar no algoritmo do método proposto e fazer a alteração de retirar o mecanismo de reparação, continuando, no entanto, a serem aceites indivíduos inválidos. Como foi já referido no *Capítulo 3*, houve já algumas abordagens com esta técnica (veja-se, por exemplo, [Rat04]), no entanto apenas como suporte a outros mecanismos; e interessava conhecer também quais as vantagens da utilização do mecanismo de reparação, *versus* a não utilização, utilizando exactamente os mesmos parâmetros de configuração do algoritmo.

Assim, da comparação entre os resultados da utilização do primeiro conjunto de parâmetros (consulte-se a Tab. 5.1) nas execuções das sequências de teste padrão, com e sem aplicação do mecanismo de reparação (ver Tabs. 5.10 e 5.12, respectivamente), pode-se inferir que o mecanismo de reparação tem efeitos positivos sobre a evolução da população. Senão, repare-se que com a aplicação do mecanismo de reparação se conseguiram igualar, logo à partida, nove resultados e, sem o mecanismo de reparação, apenas sete resultados. Para além disso, também é possível observar que os resultados médios são inferiores sem a aplicação do mecanismo, ao passo que os desvios padrão aumenta.

Também comparando-se os melhores resultados apresentados na Tab. 5.4 com a coluna *GA* da Tab. 5.9, pode-se reparar que a abordagem sem o mecanismo de reparação e sem aceitar indivíduos inválidos apresenta resultados bastante semelhantes à abordagem de Unger e Moult [UM93], havendo uma sequência (a quarta) onde é melhor e duas onde é pior (a sétima e a oitava). Contudo, crê-se que, nestas duas sequências, as mais longas testadas em [UM93], os resultados poderiam ser igualados ou superados, se se tivesse aumentado o espaço de procura (aplicando, por exemplo, os restantes conjuntos de parâmetros), tal como foi feito com as outras execuções

onde o mecanismo de reparação foi utilizado. Estabelece-se assim que os resultados deste algoritmo (sem reparação e sem indivíduos inválidos) seriam pelo menos tão bons como os de uma abordagem com um algoritmo genético clássico.

Interessa agora comparar, na abordagem apresentada, os resultados com e sem aplicação do mecanismo de reparação, e também com a aceitação ou não de indivíduos inválidos. Note-se que, para efectuar esta comparação, apenas foi estudado o primeiro conjunto de parâmetros apresentado na Tab. 5.1. Comparando-se as Tabs. 5.3, 5.4 e 5.5, com os resultados com reparação, sem reparação e sem indivíduos inválidos, e só sem reparação, respectivamente, pode-se logo à partida observar que os resultados com reparação são substancialmente melhores: não só foram igualados nove resultados ao invés de sete ou seis, como, de forma mais evidente, os resultados médios são bastante superiores (por vezes na casa das unidades), bem como os desvios padrão são notoriamente menores. Assim, olhando apenas para os resultados, pode-se inferir que existe vantagem na utilização do mecanismo de reparação. Comparando-se apenas os resultados presentes nas Tabs. 5.4 e 5.5, é possível também observar que a aceitação de indivíduos inválidos na população traz benefícios. Pode-se concluir que a aceitação de indivíduos inválidos facilita a evolução da população e também que a reparação (de parte) desses mesmo indivíduos também traz benefícios evidentes a qualidade geral da população.

Uma observação óbvia, a partir da Tab. 5.10, é que, de facto, existem bastantes indivíduos inválidos — o que não surpreende de todo, já que com cadeias a partir de quatro aminoácidos é bastante provável a ocorrência de conformações em que há aminoácidos sobrepostos. Sabendo-se à partida que a dimensão da descendência seria 675 ($500 \times 1{,}35$, ver Tab. 5.1), uma percentagem compreendia entre os 33% (para as sequências mais pequenas) e os 72% (para as sequências maiores) correspondia a indivíduos inválidos, que posteriormente foram sujeitos a reparação.

Para as execuções com os restantes conjuntos de parâmetros, o valor máximo para os indivíduos sujeitos a reparação também é semelhante, situando-se nos 65%, para os *conjuntos 2* e *3*, e nos 68%, para o *conjunto 4*. Assim, se tivermos em conta que nem todos os indivíduos seriam reparados (devido ao patamar definido para acesso à reparação), o número de indivíduos inválidos criados em cada descendência é realmente bastante elevado.

Tendo em conta a quantidade de indivíduos inválidos, restavam três hipóteses para lidar com o problema:

- a mais comum, e presente em abordagens descritas com maior pormenor no *Capítulo 3*, era pura e simplesmente descartar esses mesmos indivíduos, continuando-se a aplicar os operadores de variação até se obterem indivíduos válidos ou, se tal não fosse o caso, criar réplicas dos progenitores (como foi aqui experimentado quando não se utilizou o mecanismo de reparação);

- outra abordagem era penalizar os indivíduos inválidos e deixá-los no meio da população (penalizando-os de alguma forma), tendo-se o cuidado de não os deixar ser uma solução candidata a melhor indivíduo, criando diversidade, mas não garantindo *a priori* que a qualidade média dos indivíduos se eleve;

- finalmente, outra abordagem possível — a aqui aplicada e, tanto quanto se sabe, inédita — era procurar reparar os indivíduos inválidos, garantindo-se diversidade e, ao mesmo tempo, elevando a qualidade média da população.

Desta forma, adoptando-se a terceira hipótese, para além das vantagens descritas, há outra interessante: em sequências com um tamanho na ordem dos 50 aminoácidos, ou superior, tendo-se um indivíduo inválido, após a aplicação de um qualquer operador de variação (seja ele a recombinação, a mutação ou a macromutação), ou mesmo na sua geração, é mais célere repará-lo do que continuar indefinidamente com a criação de indivíduos inválidos até que um válido seja criado — e isto será tanto mais evidente quanto maior forem as sequências de aminoácidos. Ou seja, o custo computacional da adopção do mecanismo de reparação será inferior ao da não adopção do mesmo mecanismo, em especial nas sequências mais longas.

Para corroborar a afirmação de que a reparação de um indivíduo é muito menos dispendiosa, em termos de tempo, que aguardar a criação de um indivíduo válido, excepto em sequências pequenas, foram realizados dois conjuntos de testes onde foi replicada apenas a criação de uma população inicial de 500 indivíduos para todas as sequência de teste padrão. No primeiro conjunto de testes (ver Tab. 5.16) foram usadas as definições padrão da abordagem proposta, com reparação e aceitação de indivíduos inválidos. No segundo conjunto de testes (ver Tab. 5.17) não foram aceites indivíduos inválidos nem foi efectuada qualquer reparação. Em ambas as tabelas são apresentados os valores médios de 30 execuções para cada uma das sequências de teste padrão.

Comparando as duas tabelas, é possível observar que apesar da adopção do mecanismo de reparação e da aceitação de indivíduos inválidos na população tomar mais

Tabela 5.16: Tempo médio tomado para criar uma população, com reparação e com indivíduos inválidos

Seq.	Tempo (ms)	Indivíduos Inválidos	Reparações Tentadas	Reparações com Sucesso
1	73,87	391,13	278,50	278,50
2	45,00	430,30	312,07	312,07
3	43,60	440,13	317,67	317,67
4	102,90	482,93	376,83	376,83
5	225,60	496,37	412,93	412,93
6	519,73	496,77	421,43	421,43
7	992,93	499,03	439,67	439,67
8	1265,27	499,47	447,60	447,60
9	69,93	394,53	282,50	282,50
10	2410,80	499,97	469,57	469,57
11	7123,20	500,00	431,53	431,53
12	8532,93	500,00	431,90	431,90
13	41,97	366,53	258,77	258,77
14	41,93	363,43	254,83	254,83
15	38,67	366,73	259,80	259,80

tempo que a não aplicação da reparação e não aceitação de indivíduos inválidos nas sequências mais pequenas — as sequências 1, 2, 3, 4, 9, 13, 14 e 15 —, nas outras sequências é bastante mais rápida. Ou seja, perde-se algum tempo nas sequências mais pequenas — no pior caso, na sequência 9, a abordagem proposta é cerca de quatro vezes mais lenta —, mas nas sequências mais longas o ganho é muito superior — na sequência 11, a abordagem proposta chega a ser mais de sessenta vezes mais rápida. Assim, o tempo ganho nas sequências maiores compensa plenamente o tempo perdido nas sequências mais curtas, justificando, pelo menos em termos de eficiência, a adopção do mecanismo de reparação.

Observando os restantes dados das tabelas, podem ser tiradas ainda tirar outras conclusões interessantes. Na Tab. 5.17 é também possível notar que nas sequências mais longas, como é o caso da 11 e da 12, que têm 100 aminoácidos, por cada indivíduo válido são criados quase 80 mil indivíduos inválidos — tem-se um total de quase 40 milhões de indivíduos inválidos para somente 500 indivíduos válidos. Com base nestes valores é fácil apontar a razão de ser da complexidade deste problema, e é também possível observar que, de facto, o espaço de conformações é muito mais povoado por conformações inválidas do que por conformações válidas. Assim, observando-se novamente a Tab. 5.16, não é surpresa que a grande maioria (ou mesmo todos, em

Tabela 5.17: Tempo médio tomado para criar uma população, sem reparação e sem indivíduos inválidos

Seq.	**Tempo (ms)**	**Indivíduos Gerados**
1	19,73	2303,83
2	24,50	3636,70
3	19,83	4071,67
4	87,90	14518,17
5	367,33	61659,33
6	1039,53	79623,23
7	3681,50	270453,47
8	5787,77	443779,63
9	17,20	2327,23
10	64592,37	6112896,83
11	434351,57	39203442,63
12	404040,13	38784481,97
13	14,70	1853,20
14	18,07	1840,37
15	21,33	1848,33

algumas das sequências mais longas) dos indivíduos de cada população tenha sido inválida.

Outro valor interessante é a taxa efectiva de reparação, obtida através do *ratio* entre os indivíduos reparados com sucesso e os indivíduos reparados (na sua totalidade). De facto, todos os indivíduos que foram sujeitos a reparação — na população inicial, só os indivíduos inválidos com uma energia mínima de conformação superior a zero seriam reparados — foram efectivamente reparados. Tais valores são um indicativo da robustez do mecanismo de reparação proposto.

Crê-se também que a análise das tabelas referentes à aplicação ou não do mecanismo de reparação durante execuções completas do algoritmo proposto (Tabs. 5.3 e 5.5, respectivamente) vem corroborar as afirmações feitas nos parágrafos anteriores, constatando-se a eficiência e necessidade de um mecanismo de reparação.

Capítulo 6

Conclusões e Trabalho Futuro

Neste final capítulo são apresentadas conclusões sobre o trabalho realizado e resultados obtidos, bem como o delinear do trabalho futuro.

6.1 Conclusões

O objectivo do trabalho era mostrar a aplicabilidade da abordagem evolucionária à previsão da conformação tridimensional de proteínas. Pelas razões expostas ao longo do texto, optou-se por se utilizar o modelo HP bidimensional, ao qual se procurou acrescentar algum novo mecanismo. Foi feito, com esse intuito, um estudo experimental extenso. É altura de sistematizar e resumir as conclusões a que se pôde chegar. Nas conclusões são realçados, essencialmente, alguns dos dados que foi possível inferir a partir da análise dos resultados, presente no *Capítulo 5*.

Nesta dissertação, a primeira conclusão que se pode tirar é a de que o mecanismo de reparação oferece um conjunto de potencialidades ao nível da validação de indivíduos: não só é mais rápido reparar um indivíduo do que esperar que um indivíduo válido seja gerado (especialmente para sequências de aminoácidos mais longas), como é melhor, em termos de diversidade e qualidade, do que descartar indivíduos inválidos e fazer *clones* dos progenitores (quando "falha" a recombinação) ou dos próprios indivíduos na etapa anterior à mutação.

É possível também observar que o mecanismo de reparação é suficientemente autónomo, não dependendo da sua inserção num algoritmo genético para poder ser aplicado. Basta que lhe seja passado um indivíduo inválido para que este, na maior parte dos casos, devolva um indivíduo reparado e válido. Será assim possível adoptá-lo para outras abordagens, mesmo não evolucionárias.

Outro aspecto interessante é o da utilização de indivíduos inválidos. Apesar de não ser algo inédito, curiosamente é algo que nunca fez a transição para uma adopção generalizada, talvez por se crer que estes nunca serão uma ajuda para a descoberta da solução óptima (necessariamente válida). No entanto, os indivíduos inválidos contribuem para uma maior diversidade da população: senão veja-se, numa proteína — e quanto mais longa, mais evidente é — o número de conformações válidas é apenas uma pequena fracção do espaço de todas as conformações possíveis. E certamente haverá um número elevado de indivíduos na *fronteira* existente entre as conformações válidas e as inválidas. Não seria interessante manter estes indivíduos, na expectativa de que, com alguma evolução, estes dêem origem (ou se "transformem") em indivíduos válidos? É essa a abordagem aqui adoptada, tendo-se o cuidado de reparar os indivíduos que não apresentam esta possibilidade de evolução, usando-se para isso o mecanismo de reparação.

Observa-se também que técnicas que mostraram resultados positivos noutros domínios demonstram também aplicabilidade neste problema específico, tais como a utilização de taxas variáveis de variação, evitando uma estagnação precoce da população e facilitando, sempre que necessário, a criação de novos indivíduos bastantes distintos dos seus progenitores, agitando o "caldeirão genético". É curioso notar que é nestas circunstâncias que o mecanismo de reparação é mais útil, já que quando a população começa a estagnar é porque ficou encurralada num mínimo local e passa a ser adoptada uma conformação única — i.e., os indivíduos passam a ser idênticos. Depois, quando esta conformação é alterada "bruscamente", através do aumento das taxas de mutação e macromutação, a probabilidade de os novos indivíduos serem inválidos passa a ser bastante superior.

Finalmente, falta referir a importância dos operadores de macromutação, que provam ser menos destrutivos que a mutação simples, como foi já evidenciado no *Capítulo 5*. Tal justifica-se pelo facto de, em conformações mais complexas, uma simples mutação poder alterar de forma bastante perniciosa o fenótipo de um indivíduo. Já os operadores de macromutação operam de maneira mais localizada, levando a que as alterações sejam feitas de forma local, proporcionando melhoramentos localizados sem influenciar negativamente a conformação geral do indivíduo.

Concluindo, há a convicção que os objectivos sobre os quais assentou este trabalho foram atingidos:

- foi apresentada uma nova abordagem evolucionária ao problema da previsão da conformação de proteínas, com resultados promissores;

- comprovaram-se os benefícios da utilização de taxas dinâmicas de variação (mantendo a diversidade da população durante um maior intervalo de tempo);

- a utilização de mecanismos de macromutação (proporcionando alterações ao genótipo de forma localizada) revela-se menos destrutiva que o comum operador de mutação, podendo levar a um estudo aprofundado no futuro;

- e o mecanismo de reparação revelou ter um impacto considerável na *qualidade* dos indivíduos da população (sejam gerados, mutados ou recombinados), proporcionando um melhor aproveitamento (de parte) do genótipo dos indivíduos inválidos, ao mesmo tempo que torna mais eficiente a obtenção de novos indivíduos.

6.2 Trabalho Futuro

Pelas afirmações contidas em [LD89], onde é demonstrado que as propriedades evidenciadas no modelo HP 2D podem ser transpostas para o modelo HP 3D, poder-se-ia pensar que não há grande vantagem em fazer a transição para o último. No entanto, apesar disso, o modelo 3D sempre se encontra mais próximo da situação real ou, pelo menos, de modelos mais realistas, podendo-se ter uma ideia mais precisa do que será a conformação de uma dada proteína. Dito de outro modo, o modelo bidimensional providencia uma maneira de validar uma abordagem e obter algumas características da proteína, mas o modelo tridimensional oferece-nos algo mais próximo do aspecto real da proteína.

Outro factor que dá um incentivo no sentido da transição é o facto de haver outras abordagens que o fizeram com sucesso, como as de Shmygelska e Hoos, cuja transição, de [SH03] para [SH05], validou a abordagem por optimização com colónia de formigas também para o modelo tridimensional, apresentando resultados promissores. Há assim a convicção que tal também é perfeitamente exequível com a abordagem aqui descrita.

Crê-se também que, com os resultados aqui obtidos e a aprendizagem daí resultante, se consiga melhorar o método, quer em termos do mecanismo de reparação em si, como de outros elementos auxiliares, como uma seriação mais específica dos operadores de macromutação.

Assim, outro aspecto que interessará abordar num possível trabalho futuro é o de um estudo mais aprofundado dos operadores de macromutação. Um ponto que levanta algumas questões é o facto de os operadores de macromutação sem busca de padrões serem, à partida, menos destrutivos que os operadores com busca de padrões — pelo menos a sua aplicação era superior nas execuções onde não eram aceites indivíduos inválidos. Já o facto de os operadores sem busca de padrões terem uma maior aplicabilidade não é motivo de surpresa, já que os operadores com busca de padrões só são aplicados se o seu padrão específico for encontrado no genótipo do indivíduo.

Em conclusão, apesar de os resultados encontrados serem satisfatórios, existe ainda bastante margem de progressão neste método onde, ao invés de se descartarem indivíduos inválidos, ou de penalizá-los severamente, ou ainda estar um tempo indeterminado à espera do próximo indivíduo válido, se procura reparar os indivíduos inválidos de uma forma rápida e eficiente.

Bibliografia

[Anf59] C. Anfinsen. *The Molecular Basis of Evolution*. John Wiley & Sons, New York, 1959.

[Anf73] C. Anfinsen. *Les Prix Nobel en 1972*, chapter Studies on the Principles that Govern the Folding of Protein Chains, pages 103–119. Nobel Foundation, Stockholm, 1973.

[BFG+98] U. Bastolla, H. Frauenkron, E. Gerstner, P. Grassberger, and W. Nadler. Testing a New Monte Carlo Algorithm for Protein Folding. *Proteins: Structure, Function, and Genetics*, 31:52–66, 1998.

[BL98] B. Berger and T. Leighton. Protein Folding in the Hydrophobic-Hydrophilic (HP) Model is NP-Complete. *Journal of Computational Biology*, 5(1):27–40, 1998.

[BS05] T. N. Bui and G. Sundarraj. An Efficient Genetic Algorithm for Predicting Protein Tertiary Structures in the 2D HP Model. In H.-G. Beyer and U.-M. O'Reilly, editors, *GECCO 2005: Proceedings of the 2005 conference on Genetic and evolutionary computation*, pages 385–392. ACM Press, 2005.

[CDK03] V. Chandru, A. DattaSharma, and V. S. A. Kumar. The algorithmics of folding proteins on lattices. *Discrete Applied Mathematics*, 127(1):145–161, 2003.

[CN03] J.-M. Claverie and C. Notredame. *Bioinformatics for Dummies*. Wiley Publishing, New York, 2003.

[CS04] E. Costa and A. Simões. *Inteligência Artificial — Fundamentos e Aplicações*. FCA, Lisboa, 2004.

[Dar05] C. Darwin. *A Origem das Espécies*. Publicações Europa-América, Mem Martins, 2005. Translation: Dora Batista.

[Daw06] R. Dawkins. *The Selfish Gene*. Oxford University Press, Oxford, 30th anniversary edition, 2006.

[Dil90] K. A. Dill. Dominant Forces in Protein Folding. *Biochemistry*, 29(31):7133–7155, August 1990.

[DMC96] M. Dorigo, V. Maniezzo, and A. Colorni. The Ant System: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics — Part B*, 26(1):1–13, 1996.

[Glo90] L. Glover. Tabu Search: A Tutorial. *Interfaces*, 20(4):74–94, 1990.

[GS03] G. W. Greenwood and J.-M. Shin. *Evolutionary Computation in Bioinformatics*, chapter On the Evolutionary Search for Solutions to the Protein Folding Problem, pages 115–136. Morgan Kaufmann, San Francisco, 2003.

[HI95] W. Hart and S. Istrail. Fast Protein Folding in the Hydrophobic-Hydrophilic Model Within Three-eights of Optimal. In *Proceedings of the Twenty-seventh Annual ACM Symposium on Theory of Computing*, pages 157–168. ACM Press, 1995.

[Hol92] J. H. Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. MIT Press, Massachusetts, 1992.

[HYTY05] Y.-Y. Huang, C.-B. Yang, K.-T. Tseng, and C.-N. Yang. Protein Folding Prediction with Genetic Algorithms. In *Proceedings of the 4th Conference on Information Technology and Applications in Outlying Islands*, pages 130–139, 2005.

[JCSM03] T. Jiang, Q. Cui, G. Shi, and S. Ma. Protein Folding Simulations of the Hydrophobic-Hydrophilic Model by Combining Tabu Search with Genetic Algorithms. *Journal of Chemical Physics*, 119(8):4592–4595, 2003.

[KBBH02] N. Krasnogor, B. Blackburne, E. Burke, and J. Hirst. Multimeme Algorithms for Protein Structure Prediction. In *Proceedings of the International Conference on Parallel Problem Solving from Nature VII*, pages 769–778. Springer, 2002.

[KHSP99] N. Krasnogor, W. Hart, J. Smith, and D. Pelta. Protein Structure Prediction with Evolutionary Algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1596–1601. Morgan Kaufmann, 1999.

[KS05] N. Krasnogor and J.E. Smith. A Tutorial for Competent Memetic Algorithms: Model, Taxonomy and Design Issues. *IEEE Transactions on Evolutionary Computation*, 9(5):974–488, 2005.

[LD89] K. F. Lau and K. A. Dill. A Lattice Statistical Mechanics Model of the Conformational and Sequence Spaces of Proteins. *Macromolecules*, 22:3986–3997, 1989.

[LW01] F. Liang and W. H. Wong. Evolutionary Monte Carlo for Protein Folding Simulations. *Journal of Chemical Physics*, 115(7):3374–3380, 2001.

[Men96] G. Mendel. Experiments in Plant Hybridization. In *Verhandlungen des naturforschenden Vereines in Brünn, Bd. IV für das Jahr 1865*, pages 3–47. Electronic Scholarly Publishing, 1996.

[Mos89] P. Moscato. On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms. Technical Report C3P 826, California Institute of Technology, California, 1989.

[MRRT53] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, and A. H. Teller. Equation of State Calculations by Fast Computing Machines. *Journal of Chemical Physics*, 21(6):1087–1092, 1953.

[NM92] J.T. Ngo and J. Marks. Computational Complexity of a Problem in Molecular-Structure Prediction. *Protein Engineering*, 5(4):313–321, 1992.

[Pel02] D. Pelta. *Algoritmos Heurísticos en Bioinformática*. PhD thesis, Universidad de Granada, Granada, Noviembre 2002.

[PPG95] A. L. Patton, W. F. Punch III, and E. D. Goodman. A Standard GA Approach to Native Protein Conformation Prediction. In Larry Eshelman, editor, *Proceedings of the 6th International Conference on Genetic Algorithms*, pages 574–581. Morgan Kaufmann, 1995.

[Rat04] V. Ratakonda. Protein Structure Prediction. Project Report. Unpublished. Available online: `http://www.bridgeport.edu/sed/projects/cs597/Summer2003/projectreport.doc`, April 2004.

[RCMJJ04] G. J. Rylance, G. A. Cox, T. V. Mortimer-Jones, and R. L . Johnston. A Genetic Algorithm for the Investigation of Simple Protein Folding Models. Poster. Unpublished. Available online: `http://tc.bham.ac.uk/~gjr/Refs/Bonn_poster.pdf`, 2004.

[Ric91] F. M. Richards. The Protein Folding Problem. *Scientific American*, 264(7):54–63, 1991.

[Sch06] R. Schleif. Analysis of Protein Structure and Function: A Beginner's Guide to CHARMM. Unpublished. Available online: `http://gene.bio.jhu.edu/book.pdf`, January 2006.

[Sea00] D. B. Searls. Using Bioinformatics in Gene and Drug Discovery. *Drug Discovery Today*, 5(4):135–143, 2000.

[SH03] A. Shmygelska and H. Hoos. An Improved Ant Colony Optimisation Algorithm for the 2D HP Protein Folding Problem. In *Proceedings of the 16th Conference of the Canadian Society for Computational Studies of Intelligence*, pages 400–417. Springer-Verlag, 2003.

[SH05] A. Shmygelska and H. Hoos. An Ant Colony Optimisation Algorithm for the 2D and 3D Hydrophobic Polar Protein Folding Problem. *BMC Bioinformatics*, 6:30, 2005.

[Sil99] S. Silva. Previsão da estrutura secundária de proteínas utilizando redes neuronais. Master's thesis, Universidade de Lisboa, Lisboa, Outubro 1999.

[SK03] S. Schulze-Kremer. *Evolutionary Computation in Bioinformatics*, chapter Application of Evolutionary Computation to Protein Folding with Specialized Operators. Morgan Kaufmann, San Francisco, 2003.

[SL02] R. Samudrala and M. Levitt. A Comprehensive Analysis of 40 Blind Protein Structure Predictions. *BMC Structural Biology*, August 2002.

[Spe93] W. M. Spears. Crossover or Mutation? In L. D. Whitley, editor, *Proceedings of the Second Workshop on Foundations of Genetic Algorithms*, pages 221–237, California, 1993. Morgan Kaufman.

[Str95] L. Stryer. *Biochemistry*. W. H. Freeman & Company, New York, 1995.

[UM93] R. Unger and J. Moult. Genetic Algorithms for Protein Folding Simulations. *Journal of Molecular Biology*, 231:75–81, 1993.

[WD83] C. Wuilmart and P. Delhaise. Linear Repetitions of Amino Acids and Convergent Evolution Inside Protein Subregions of Ordered Secondary Structures. *Journal of Molecular Evolution*, 19:355–361, 1983.