UM ALGORITMO HÍBRIDO PARA EXTRAÇÃO DE CONHECIMENTO EM BIOINFORMÁTICA

Ricardo Linden

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Aprovada por:

_____

Prof. Amit Bhaya, Ph.D.

_____

Prof. Alexandre Gonçalves Evsukoff, Dr.

_____

Profa. Ana Lúcia Cetertich Bazzan, Dr. Ing.

_____

Prof. André Carlos Ponce de Leon de Carvalho, Ph.D.

_____

Prof. Nelson Francisco Favilla Ebecken, D. Sc.

RIO DE JANEIRO, RJ – BRASIL

ABRIL DE 2005

LINDEN, RICARDO

Um Algoritmo Híbrido Para Extração de Conhecimento em Bioinformática [Rio de Janeiro] 2005

IX, 204p, 29,7 cm (COPPE/UFRJ, D.Sc., Engenharia Elétrica, 2005)

Tese – Universidade Federal do Rio de Janeiro, COPPE

1. Bioinformática

2. Classificação

3. Programação Genética

4. Lógica Nebulosa

I. COPPE/UFRJ II. Título (série)

Dedicatória

Gostaria de dedicar esta tese à minha esposa Claudia Wolff, que foi minha companheira em todos os momentos da elaboração desta tese, tanto os bons e ruins, e sofreu todas as dores do parto comigo.

Gostaria também de dedicar esta tese a meus pais, que sempre me incentivaram, e que me impediram de tentar ser jogador de futebol profissional.

Não poderia me esquecer de dedicar esta tese também aos meus amigos da D-11. Paulo, Fátima, Marcelo, Guilherme, Bianco e Plutarcho foram sempre companheiros nesta longa jornada.

Agradecimentos

A realização desta tese não seria possível sem a ajuda e compreensão de duas pessoas muito especiais.

A primeira é o meu amigo e chefe no CEPEL, Victor Navarro Araújo Lemos da Silva, que sempre ofereceu sua amizade quando mais necessitei durante todo este período, compreendendo todas as dificuldades impostas por um doutorado e ajudando sempre que necessário.

A segunda é a reitora da Faculdade Salesiana Maria Auxiliadora, onde leciono, a Irmã Maria Léa Ramos. A Irmã Léa criou um ambiente agradável e amistoso, onde é ótimo trabalhar e sempre incentivou a todos os professores, e a mim especialmente, a procurar crescer como profissionais.

Além disto, gostaria de agradecer ao Dr. Marcos Antônio dos Santos, que gastou seu tempo explicando conceitos de oncologia e analisando os conjuntos de dados de sua área, dando maior validade à análise realizada nesta tese.

A todos, meu mais sincero obrigado.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc)

UM ALGORITMO HÍBRIDO PARA EXTRAÇÃO DE CONHECIMENTO EM BIOINFORMÁTICA

Ricardo Linden

Abril/2005

Orientador: Amit Bhaya

Programa: Engenharia Elétrica

Nesta tese, lógica nebulosa, algoritmos evolucionários, e um algoritmo iterativo para agrupamento de dados categóricos foram usados para propor um novo algoritmo para analisar vários problemas para os quais existem dados insuficientes para a aplicação de técnicas estatísticas tradicionais. Este algoritmo permite que sejam tratados simultaneamente dados Booleanos, contínuos e categóricos ou não numéricos.

Este algoritmo foi utilizado em dois tipos de aplicações principais: engenharia reversa de redes de regulação genética e classificação de dados numéricos e categóricos.

Os problemas de bioinformática analisados consistem em extração de relacionamentos regulatórios a partir de uma quantidade insuficiente de dados para permitir uma análise utilizando técnicas convencionais, como estatística. Neste contexto, obtêm-se resultados que servem como guias de pesquisa, permitindo que se façam menos experimentos laboratoriais, efetivamente direcionados para a obtenção de reguladores reais.

No caso de problemas de classificação, o algoritmo proposto foi aplicado em alguns dos problemas mais utilizados como *benchmark* no meio de classificação e mostrou-se capaz de obter resultados comparáveis aos melhores métodos existentes, porém com maior consistência e interpretabilidade para o usuário final.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc)

A HYBRID ALGORITHM FOR KNOWLEDGE EXTRACTION IN BIOINFORMATICS

Ricardo Linden

April/2005

Advisor: Amit Bhaya

Department: Electrical Engineering

In this work fuzzy logic, evolutionary algorithms and an iterative algorithm to cluster categorical data are used to develop a new algorithm to analyze several problems for which there is insufficient data to allow the application of traditional statistical techniques. This algorithm allows for simultaneous usage of Boolean, numerical and categorical data.

This algorithm was used in two main applications: reverse engineering of regulation networks and categorical and numerical data clustering.

The bioinformatics problem studied in this thesis is the extraction of regulatory relationships from a data set, which is too small for the application of conventional techniques such as statistical analysis. In this scenario, using the proposed approach, it is possible to obtain results that serve as research guidelines, allowing the realization of biological lab experiments that are more effectively directed to the finding of the real regulators.

For classification problems, the proposed algorithm was applied to several problems that have traditionally been considered as benchmarks in the area. It has consistently achieved results that are comparable to the best methods, but in a format that is more comprehensible for the final user.

Índice

Tabela de Símbolos

$b_{vik}(t)$	valor assumido pelo k-ésimo gene regulador do gene i no instante t.
EA	Algoritmo Evolucionário
GA	Algoritmo Genético
GP	Programa Genético
H(Y|X)	entropia da variável Y condicional à variável X
$H(Y\|X=\upsilon_k)$	entropia condicional específica da variável Y quando X assume o valor dado por υ_k
k	número de genes reguladores de um nó
K	número de arestas saindo de um nó de uma rede Booleana com número fixo de reguladores.
MAPE	Mean Absolute Percent Error – média dos módulos dos erros percentuais cometidos
n	número de genes presentes nos dados
SVM	Support Vector Machine
$u_{\tau i}$	valor da i-ésima coluna (variável) de uma tupla τ
v_{jn}	n-ésimo gene regulador do gene j, onde n=1,...,k.
$w_{u_{\tau i}}$	peso associado ao valor $u_{\tau i}$ durante a execução do STIRR
ρ_{ij}	correlação entre o padrão de expressão dos genes i e j.
τ	uma tupla de um conjunto de dados.
υ_k	um dos n valores distintos assumidos por uma variável em um conjunto de dados
Ω	Limite assintótico mínimo da quantidade de tempo ou dados necessários para um algoritmo.

Capítulo 1 Introdução

Neste capítulo serão discutidas brevemente as motivações deste trabalho, apresentando o que já foi feito anteriormente nesta área e as principais melhorias alcançadas pelo trabalho desenvolvido. Todos os conceitos descritos neste capítulo serão discutidos de forma mais detalhada nos capítulos a seguir.

1.1 Objetivos deste trabalho

O objetivo inicial deste trabalho foi desenvolver uma ferramenta capaz de analisar dados escassos, provenientes de fontes de difícil acesso, tanto por questões de ordem prática quanto financeira. Estas ferramentas se mostravam necessárias para analisar conjuntos de dados cuja dimensionalidade não permitiria a aplicação de técnicas tradicionais de análise estatística.

As técnicas tradicionais necessitam de conjuntos volumosos de dados, com tamanhos "estatisticamente significativos". Entretanto, existem várias considerações experimentais que podem impedir a obtenção de tais conjuntos. Um exemplo claro ocorre na área de análise de dados de microarrays, na qual cada experimento pode custar centenas, ou mesmo milhares de dólares, o que limita sua reprodutibilidade em laboratórios que não disponham de recursos financeiros deste porte. Isto não deveria impedir que estes dados

fossem analisados, mesmo que os resultados desta análise sejam vistos com cautela (por falta de significância estatística).

Tudo isto nos impeliu para a criação de nosso algoritmo e da ferramenta que o implementa na prática. Esta ferramenta serve como uma plataforma de análise de dados bem como uma geradora de hipóteses que devem ser analisadas a posteriori em condições reais (como a bancada de um laboratório de análises genéticas, por exemplo).

Estas hipóteses, apesar de não serem estatisticamente significativas, servem como guias de pesquisa, fornecendo estradas promissoras para serem trilhadas por cientistas que desejarem realizar suas pesquisas de forma mais eficiente.

No decorrer do trabalho, percebemos que o algoritmo proposto é eficiente na extração de conhecimento subjacente aos dados, processo este conhecido como mineração de dados. Assim, ampliamos nossos esforços para contemplar esta importante área da computação, mais especificamente, para a tarefa de classificação de conjuntos de dados.

Estes dois objetivos não são mutuamente excludentes, mas sim demonstram a possibilidade de se aplicar o trabalho aqui desenvolvido em áreas correlatas nas quais há necessidade de novas ferramentas com as características descritas acima..

1.2 Revisão Bibliográfica

A idéia de utilizar algoritmos evolucionários para desenvolver bases de regras fuzzy não é nova, e tem sido explorada intensamente na literatura.

HERRERA *et al.* (1997) foram um dos primeiros a descrever detalhadamente o uso de algoritmos genéticos na evolução de sistemas fuzzy. Este trabalho consiste em um tutorial que descreve as principais características do desenvolvimento de regras fuzzy usando algoritmos genéticos. Este trabalho contém várias descrições de como evoluir bases de regras, algumas das quais foram aproveitadas nos capítulo 3 e 4 desta tese.

(FREITAS, 2000) faz uma revisão da literatura em que descreve os principais usos de algoritmos genéticos na área de mineração de dados, na qual o trabalho descrito nesta

tese se encaixa. Este artigo descreve vários modelos de busca de conhecimento, isto é, formas diferentes de evoluir de bases de regras.

Alguns artigos recentes utilizaram algoritmos evolucionários para o desenvolvimento de bases de regra fuzzy. Entre eles podemos citar (BENTLEY, 1999, DASGUPTA *et al.*, 2002, MENDES *et al.*, 2001, HETLAND *et al.*, 2002. Todos eles têm características particulares que são comparadas com a representação utilizada no capítulo 4 desta tese.

A busca de redes de regulação de DNA, uma das aplicações discutidas neste trabalho, tem sido uma grande preocupação recente da ciência. DE JONG (2002) apresenta uma extensa revisão do trabalho realizado nesta área. Entre os trabalhos citados nesta revisão, pode-se destacar aquele apresentado por SMOLEN *et al.* (2000), que usa modelos matemáticos baseados em equações diferenciais e o de WAGNER (2001), que se baseia na análise de perturbação de redes. Ambos os modelos foram descartados por se basearem em uma quantidade de dados que atualmente não é facilmente atingível, tornando-os inviáveis para as aplicações alvo deste trabalho.

Existem trabalhos baseados em análise perturbacional que não necessitam de dados pós-perturbação de todos os genes em consideração, tais como (SONTAG *et al.*, 2004). Entretanto, por questões estatísticas, desconsideradas em tal artigo, múltiplas perturbações deveriam ser realizadas e por conseguinte, o número de experimentos necessários ainda se mostra pouco prático devido a questões monetárias.

O problema da limitação de recursos financeiros é real e atinge praticamente todos os laboratórios que realizam pesquisa baseada em microarrays. ZIEN *et al.* (2002) busca quantificar o número de microarrays necessários, de forma a organizar e limitar uma pesquisa, obtendo uma fórmula para se calcular o número desejável de experimentos e replicações, de forma a obter um conjunto de dados o mais significativo possível.

Existem trabalhos que trazem enfoques computacionais relevantes. HAUTANIEMI (2003), por exemplo, apresenta um capítulo sobre a escolha de variáveis de interesse em um processo de desenvolvimento de câncer. A seleção de variáveis é feita através do conceito de variável expressa de forma diferenciada, através de uma razão em relação à normalidade. Posteriormente, é usado um algoritmo hierárquico de agrupamento para separação dos dados.

Existem vários trabalhos na área biológica que buscam interpretar as redes de regulação genética sob uma abordagem mais sistemática, fornecendo os mecanismos básicos de compreensão das mesmas para a aplicação de técnicas computacionais como a proposta nesta tese.

DAVIDSON *et al.* (2002), por exemplo, faz uma análise de desenvolvimento e perturbação da rede de regulação genética, concentrando-se na área de desenvolvimento embrionário.

O principal passo de pré-processamento adotado na aplicação descrita nesta tese consiste em agrupar os elementos que possuam um perfil de expressão semelhante, teorizando portanto que eles estejam sob o efeito de um processo de regulação semelhante. Assim, foi necessário efetuar um estudo das técnicas de agrupamento que poderiam se mostrar úteis na aplicação proposta.

JIANG *et al.* (2002) apresentam uma revisão abrangente do uso deste processo de agrupamento (clustering) aplicado a dados biológicos, analisando desde as técnicas de obtenção de dados até os métodos mais tradicionais de agrupamento, tais como K-Means, Agrupamento Hierárquico, baseada em grafos e SOM, com suas respectivas aplicações na área de análise de dados de microarrays.

HANISCH *et al.* (2002) apresenta uma estratégia mais avançada que inclui a utilização de modelos teóricos associados aos dados na hora de se fazer o agrupamento. Esta estratégia discute a aplicação da técnica que é denominada co-clustering na análise de redes metabólicas.

Outro ponto importante a se considerar é a questão da utilização do conhecimento disponível *a priori*. SCHRAGER *et al* (2002) discutem a importância da idéia de poder incluir modelos teóricos na análise dos dados.

Este conceito está embutido na ferramenta aqui desenvolvida, que permite que o usuário inclua dados obtidos em modelos teóricos na análise que o programa efetua. Este tipo de abordagem é mais completo, tendo em vista que não despreza o conhecimento pré-existente, usando-o como uma plataforma a partir da qual torna-se mais fácil obter resultados práticos.

O uso de algoritmos genéticos na busca de redes de regulação de DNA também tem sido motivo de forte pesquisa recente. Entre os vários trabalhos disponíveis, podemos

destacar o de (SZALLASI *et al*, 1998), (AKUTSU *et al.*, 1999) e a tese de doutorado descrita em (D'HAESELEER, 1993), que desenvolveram um extenso trabalho na identificação de redes de regulação usando o modelo Booleano (descrito detalhadamente no capítulo 3 desta tese) e ANDO *et al.* (2000), que utiliza representações reais e grafos, representados usando-se matrizes.

SPIETH *et al.* (2004) adotam uma abordagem diferente para algoritmos evolucionários neste problema, definindo um modelo fechado, baseado em sistemas S (tipo especial de equações diferenciais), usando um algoritmo genético para otimizar os parâmetros destas equações.

Assim como o trabalho descrito nesta tese, muitos outros pesquisadores buscaram usar ferramentas de clustering de forma a pré-processar os dados decorrentes de microarray, buscando diminuir a quantidade de dados que estes disponibilizam ou simplesmente para obter mais informação a partir dos dados brutos. As diferenças entre estes trabalhos serão apresentadas na seção 5.8 desta tese.

Outros trabalhos interessantes em áreas correlatas fornecem informações interessantes sobre a evolução de redes de regulação. Por exemplo, KOZA *et al.* (2001) descrevem a evolução de redes de regulação metabólica usando programação genética, trabalho muito próximo, em essência, à aplicação que é discutida no capítulo 5 desta tese.

PIASECZNY *et al.* (2004) descreve um novo e interessante algoritmo evolucionário que se baseia na relação entre DNA e aminoácidos para evoluir soluções para um problema. Entretanto, tal algoritmo é bastante recente, e aplicações mais avançadas do mesmo ainda não estão disponíveis.

Como dados de microarray normalmente tem uma dimensão (quantidade de genes medidos) duas ou três ordens de grandeza maior do que a outra (quantidade de medidas), ALTER *et al.* (2001) sugerem que uma idéia razoável seria aplicar a decomposição em valores singulares, para torná-la mais tratável.

A área de classificação, aplicação descrita no apêndice A desta tese, é muito rica em trabalhos. Vários trabalhos utilizam os exemplos tipo benchmark usados nesta tese como forma de avaliar seu desempenho, entre os quais podemos destacar (EGGERMONT *et al.*, 2004a), que usa uma estratégia evolutiva para criar uma árvore de decisão,

DOUNIAS *et al.* (2002) e ZHOU *et al.* (2002) apresentam uma abordagem similar àquela adotada nesta tese, de evolução de regras para classificação. Entretanto, os operadores e regras desenvolvidas nestes trabalhos apresentam algumas diferenças significativas que os tornam menos completos do que os descritos nesta tese, como será visto no apêndice A.

Ao buscar um algoritmo que pudesse ajudar na análise de dados categóricos, encontra-se uma seleção extensa de pesquisas que descrevem algoritmos interessantes e poderosos. CHEN *et al.* (2003) apresenta uma excelente revisão do tópico, enquanto que SCHEAFFER (1999) descreve algumas técnicas estatísticas tradicionais usadas nesta área.

PALMER *et al.* (2003) apresenta um dentre vários trabalhos que buscam criar uma maneira de se medir uma distância numérica entre atributos categóricos de forma a se utilizar os algoritmos tradicionais, tais como K-Means ou hierárquicos.

Outras abordagens buscam criar algoritmos que se baseiam nos relacionamentos dos dados categóricos em todo o conjunto de dados. GUHA *et al.* (2000) e CRISTOFOR *et al.* (2000) descrevem dois algoritmos interessantes cujo uso foi descartado em favor do algoritmo STIRR (GIBSON *et al.*, 2000, ZHANG *et al*, 2000), que cria um sistema dinâmico baseado nos dados e cuja descrição detalhada encontra-se no capítulo 4 desta tese, na seção que descreve a aplicação do algoritmo a dados categóricos.

Quando se usam algoritmos evolucionários para evoluir uma base de dados fuzzy, o espaço de busca é extremamente grande. Estes algoritmos podem beneficiar-se de medidas de redução do espaço de busca, tais como as descritas em (EGGERMONT *et al.*, 2004a), sem grande perda de precisão. KIM *et al.* (2002) buscam escolher de forma mais eficiente quais variáveis merecem ser considerados na elaboração dos conjuntos fuzzy utilizando um algoritmo evolucionário na busca. Isto faz com que seja acrescentado mais um passo lento, prejudicando sobremaneira o desempenho final do algoritmo.

1.3 Contribuição desta Tese

Durante o desenvolvimento desta tese, fez-se um estudo abrangente das técnicas existentes para a mineração de dados numéricos, especialmente aquelas concentradas em fontes de dados escassas.

Fez-se também uma análise extensa das técnicas aplicáveis a dados categóricos, bem como aquelas que permitem que se lide com dados numéricos e categóricos simultaneamente.

Os trabalhos de ambas as categorias foram sujeitos a uma análise crítica, apontando deficiências, que são descritas no corpo desta tese. Buscou-se pontuar as críticas com um caráter essencialmente prático, de forma que futuros usuários, tanto dos algoritmos descritos aqui, quanto dos descritos nos artigos e teses citados, possam aprimorar suas técnicas e obter resultados ainda melhores do que aqueles citados na literatura.

Com base nestes estudos, desenvolveu-se uma ferramenta baseada em um algoritmo criado para esta tese, ferramenta esta que serve para realizar uma análise de conjunto de dados, extraindo conhecimento de fontes cujo tamanho faz com que seja impraticável aplicar técnicas estatísticas tradicionais.

Esta ferramenta pode ser usada também como forma de extrair conhecimento de conjuntos de dados que incluam tanto dados numéricos quanto categóricos, e, em princípio, não possui as deficiências levantadas acima.

Esta ferramenta busca fornecer resultados inteligíveis que podem ser facilmente compreendidos e utilizados pelos pesquisadores que dela fizerem uso. Isto é uma vantagem grande sobre várias abordagens de caixa-preta, como redes neurais e alguns métodos de clustering.

Ademais, a ferramenta proposta possui a capacidade de incorporar conhecimento prévio, diferentemente da maioria das técnicas estudadas. Esta capacidade é interessante

pois em várias áreas, especialmente na biologia, existe uma ampla gama de conhecimentos disponível que não deve ser desprezada por um pesquisador.

A ferramenta proposta aqui possui um espectro amplo de aplicações, incluindo a área de análise de microarrays e a classificação, o que é demonstrado através de exemplos tipo *benchmark* apresentados ao longo do trabalho.

Resultados preliminares deste trabalho foram apresentados nos seguintes congressos:

- XII Artificial Neural Network in Engineering (ANNIE), realizado em Saint Louis, EUA em Novembro/2002
- I Workshop Brasileiro de Bioinformática (WOB), em formato de poster, realizado em Gramado em Novembro/2002
- III BIOMAT, realizado no Rio de Janeiro, em novembro/2003

Além disto, este trabalho gerou um artigo que foi submetido a uma revista indexada, denominada *Bioscience*, artigo este que recebeu uma primeira avaliação positiva e se encontra na segunda e última fase de revisão para publicação.

Outra conseqüência dos estudos realizados nesta tese, apesar de não ser diretamente ligado ao tema proposto aqui, foi o tutorial apresentado na IV Jornada Iberoamericana de Ingeniería de Software y Ingeniería de Conocimiento, realizado em Madrid, em novembro de 2004. O tema deste tutorial foi agrupamento de dados numéricos e categóricos, e as informações para a preparação do mesmo foram obtidas dos estudos realizados para a fase de agrupamento de dados para a aplicação de bioinformática.

1.4 Organização deste Trabalho

Nesta tese será apresentado um algoritmo que combina as características da lógica fuzzy com as dos algoritmos evolucionários para analisar dados escassos, que não poderiam ser minerados de forma eficiente por técnicas estatísticas.

No capítulo 2 é feita uma revisão de todos os conceitos utilizados nesta tese, incluindo algoritmos evolucionários e lógica fuzzy (nebulosa)

No capítulo 3 será descrita uma versão preliminar do algoritmo proposto, baseada em estruturas de dados simples que são varridas por um GA e que teve sucesso em modelar valores Booleanos (digitais). O trabalho neste algoritmo serviu como forma de identificação de algumas dificuldades do problema e facilitou o desenvolvimento do algoritmo analógico descrito no capítulo 4, que pode lidar com quaisquer valores necessários e que usa lógica fuzzy associada à programação genética.

No capítulo 5 é discutida a aplicação deste algoritmo a uma área muito promissora, a de análise de dados obtidos a partir de microarrays de DNA. Como a obtenção destes dados é muito cara, técnicas que podem obter resultados interessantes a partir de dados escassos são bem vindas.

Finalmente, no capítulo 6 são colocadas algumas conclusões e apresentados alguns desdobramentos possíveis para nosso trabalho, explicitando algumas características do algoritmo proposto que podem ser melhoradas em um futuro próximo.

No apêndice A, é apresentada a aplicação do algoritmo proposto na área de classificação, com especial ênfase na área médica. Será descrita a aplicação do trabalho desta tese em alguns conjuntos tradicionalmente usados na área, mostrando os benefícios auferidos em precisão e/ou compreensibilidade dos resultados.

Capítulo 2 Conceitos Preliminares

Neste capítulo veremos alguns dos conceitos básicos envolvidos em nosso trabalho, discutindo brevemente a lógica fuzzy e os algoritmos evolucionários, para que possamos explicar de forma mais precisa o algoritmo proposto em capítulos posteriores.

2.1 Lógica Fuzzy

A lógica proposicional tradicional lida com variáveis assumindo apenas dois possíveis estados: falso e verdadeiro. Os conjuntos tradicionais são definidos utilizando apenas com a noção de pertinência absoluta (ou o elemento pertence ou não pertence ao conjunto definido) sem nenhum tipo de gradação. Por exemplo, podemos considerar o caso de todos os carros trafegando a menos de 70km/h. Os membros deste conjunto são todos os carros *i* cuja velocidade, dada por v_i, satisfaz a seguinte definição:

$\{v_i \in \Re \mid v_i < 70\}$

Se algum carro *i* estiver trafegando a menos do que 70 km/h ele pertence ao conjunto *Elem*, o que é denominado por $i \in Elem$. Caso o carro *i* esteja trafegando a 70 km/h ou

mais, ele não pertence a este conjunto, o que é dado por $i \notin Elem$. Note-se que não há nenhum tipo de gradação neste tratamento.: se o carro estiver trafegando a 69,99km/h, ele pertence a Elem, mas se ele estiver trafegando a 70 km/h, ele não pertence a Elem.

Podemos então definir uma função de pertinência $\chi_{Elem}(i)$ que é dada por:

$$\chi_{Elem}(i) = \begin{cases} 0, i \notin Elem \\ 1, i \in Elem \end{cases}$$

Em alguns casos, como no caso da representação interna de computadores esta representação é suficiente, mas no mundo real (e em grande parte das aplicações de interesse na área da engenharia) existem propriedades que são vagas, incertas ou imprecisas e, portanto, impossíveis de serem caracterizadas por predicados da lógica clássica bivalente (PEDRYCZ *et al.*, 1998), como por exemplo a determinação da pertinência de uma pessoa ao conjunto das pessoas altas.

No cotidiano é comum que uma pessoa se depare com situações em que há propriedades imprecisas, como o fato de alguém ser alto, para as quais não se possui a noção de verdadeiro/falso perfeitamente definida. Por exemplo, não é óbvio classificar um carro como estando andando rapidamente ou não, mesmo que se tenha total conhecimento de sua velocidade corrente. Se for perguntado para várias pessoas se um carro andando a 80 km/h está rápido ou não, haverá provavelmente uma resposta gradativa, incluindo uma possibilidade distinta de recebermos uma resposta do tipo “mais ou menos” ou “mais para sim do que para não”.

Esta ambuguidade é inerente à imprecisão da definição destes conjuntos e não a um eventual desconhecimento da velocidade do carro ou da altura da pessoa. As respostas são vagas porque o conceito de “carro rápido” é vago, ao contrário da medição da sua velocidade instantânea, que é absoluta.

Uma solução possível para resolver este tipo de ambigüidade seria usar lógica multivalorada, incluindo, por exemplo, uma pertinência de 0,5 para o conceito de “mais ou menos”, mas ainda precisaríamos executar procedimentos de arredondamento, já que em algum momento teríamos que fazer uma transição brusca entre duas pertinência admitidas (por exemplo, entre 0 e 0,5). Este arredondamento continua evitando que seja embutido o conceito de mudança gradual dentro do nosso sistema, pois tudo que temos, ao introduzir este valor intermediário, são duas transições bruscas em vez de uma.

Para resolver o problema das transições bruscas, pode-se optar por utilizar a lógica fuzzy. (fuzzy, em inglês, significa incerto, duvidoso ou nebuloso, que é a tradução mais adotada), que usa graus de pertinência contínuos no intervalo [0,1] ao invés de um relacionamento de verdadeiro/falso estrito como na lógica tradicional.

A lógica fuzzy é adequada para a representação tendo em vista que a maioria dos especialistas humanos possui conhecimento que é representado em termos linguísticos, de uma maneira especialmente fuzzy. Isto decorre do fato de que é simples comunicar conhecimento desta forma e pelo fato de que alguns sistemas não possuem modelagem numérica simples, mas podem ser entendidos de forma completa por meio de noções fuzzy, como por exemplo, "se a pressão estiver alta demais, abra um pouco a válvula de pressão" (WANG, 1994).

Um termo fuzzy usado de forma rotineira em nossas comunicações (como alto, baixo ou leve) é um elemento ambíguo que pode caracterizar um fenômeno impreciso ou não completamente compreendido. Estes termos fuzzy são a base da lógica fuzzy. Isto quer dizer que a lógica fuzzy está baseada em palavras e não em números, ou seja, os valores-verdade que usamos nos controladores são pertinências a conjuntos que estão fortemente associados a termos que são expressos lingüisticamente no dia a dia, como por exemplo: quente, frio, verdade, longe, perto, rápido, vagaroso, etc. Estes termos podem ser alterados através do uso de vários modificadores de predicado como por exemplo: muito, mais ou menos, pouco, bastante, meio, etc.

Estas características fazem da lógica fuzzy uma alternativa simples para representar de forma direta o conhecimento humano, restando apenas a definição formal de como operar com os conjuntos fuzzy, definição esta que é feita através da teoria dos conjuntos fuzzy.

Baseando-se nestas características, pode-se afirmar que a lógica fuzzy pode ser considerada como uma das primeiras escolhas de aplicação nas seguintes situações:

- Em sistemas muito complexos, onde é difícil desenvolver o modelo matemático.
- Para sistemas extremamente não lineares que podem ser bem explicados heuristicamente e/ou através de termos linguísticos.

Um aspecto importante da lógica fuzzy é que ela permite que incorporemos informações que são baseada em conhecimento qualitativo ou semi-qualitativo de um processo, conhecimento este que é muito comum na biologia. Um exemplo deste

conhecimento é o efeito da ATP na fosfoenolpiruvato carboxiquinase (PCK) que é bifásico, acelerando a reação em baixas concentrações e inibindo-a em altas concentrações (LEE *et al.*, 1999). Este tipo de conhecimento poderia ser modelado por duas regras similares às seguintes:

- Se Baixa_Concentração (ATP) Então Acelere_Reação(PCK)
- Se Alta_Concentração (ATP) Então Iniba_Reação (PCK)

Estes tipos de regras são muito próximos da maneira como um especialista lida com seu conhecimento, o que faz da lógica fuzzy uma ótima ferramenta para modelar o conhecimento disponível em qualquer área.

A teoria de conjuntos fuzzy permite especificar quão bem um objeto satisfaz uma descrição vaga. Isto é feito através do estabelecimento de um grau de associação que pode variar continuamente entre 0 (falso, ou ausência total de pertinência) e 1 (verdadeiro, ou totalmente pertinente). Assim, fazemos um mapeamento do valor da variável *x* para o conjunto A ($u(x) : x_A \rightarrow [0, 1]$) que significa que *x* pertence ao conjunto *A* com um valor de pertinência entre 0 (não pertence absolutamente) e 1 (pertence totalmente).

Os valores intermediários podem ser compreendidos fazendo-se uma analogia às fotografias em preto e branco. Entre os dois valores extremos, existem vários tons de cinza. Da mesma maneira, entre a pertinência total (1) e a não pertinência (0), existem vários valores possíveis de pertinência em um conjunto. Voltando ao exemplo do carro, podemos estabelecer graus de pertinência ao conjunto dos carros rápidos, dada cada uma das velocidades que ele pode assumir. Assim, um carro que esteja trafegando a 80 km/h pode receber uma pertinência de 0,7 neste conjunto, correspondendo ao conceito de "mais ou menos" citado pelo pesquisado.

Conjuntos fuzzy são aplicáveis tanto a variáveis discretas quanto contínuas. No caso de variáveis discretas, podemos definir o conjunto através da representação de todos os elementos do universo de discurso (valores que a variável pode assumir) são associados a suas pertinências, da seguinte maneira:

$$Altos(x) = \{ {}^{1,5}\!/_{0}, {}^{1,55}\!/_{0}, {}^{1,6}\!/_{0}, {}^{1,65}\!/_{0,1}, {}^{1,70}\!/_{0,3}, {}^{1,75}\!/_{0,5}, {}^{1,80}\!/_{0,7}, {}^{1,85}\!/_{1,0}, {}^{1,90}\!/_{1,0}, {}^{1,95}\!/_{1,0}, {}^{2,0}\!/_{1,0}$$

No caso, cada elemento do universo foi representado na forma $x_i / \mu(x_i)$, onde o primeiro termo representa o elemento em questão e o segundo termo representa sua pertinência ao conjunto *Altos*.

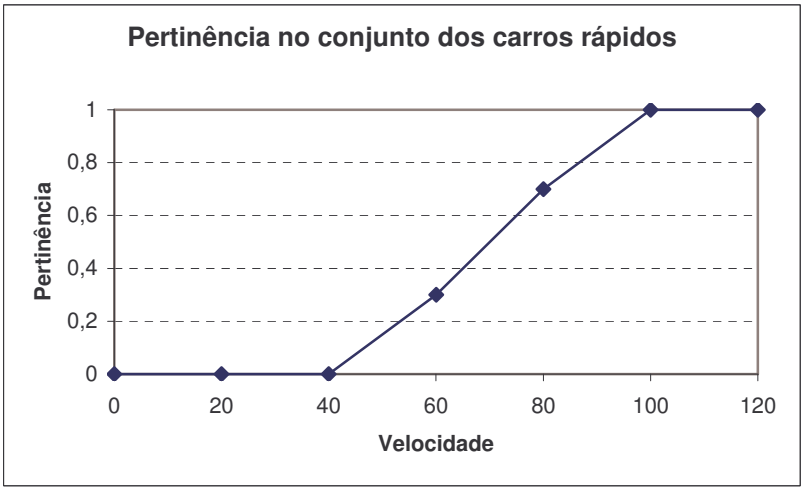


Figura 2-1: Representação de um conjunto fuzzy através do diagrama de Hassi-Euler.

Se o universo de discurso é contínuo ou possui uma quantidade grande de elementos discretos, a forma mais fácil de representação é o gráfico de sua função de pertinência, chamado de diagrama de Hassi-Euler (H-E) (EVSUKOFF *et al.*, 2003). Um exemplo de tal representação é dado na figura 2-1. Ambas as representações permitem que façamos uma associação entre o valor original da variável, que é denominado seu valor crisp, e a sua pertinência no conjunto fuzzy de interesse.

Um valor crisp é um número preciso, obtido através de um aparato medidor (velocímetro, fita métrica, etc), que representa o estado exato de um fenômeno associado e ao qual não existe nenhum tipo de definição ou ambigüidades associados, visto que este número consiste na representação de um evento físico.

A definição do conjunto fuzzy é feita por um especialista que compreende o processo a ser modelado. O conjunto fuzzy é uma representação de um termo fuzzy que extingue toda a ambigüidade associada a este termo.

A pertinência para o qual é mapeado este valor é denominado o seu valor fuzzy, valor este que não contém nenhum tipo de ambigüidade. O processo de tansformação de um valor crisp para um valor fuzzy é chamado de fuzzyficação.

Na lógica fuzzy não se afirma que um carro trafegando a 65km/h está andando rápido. Existe uma definição do que consiste andar rápido, feita através de um dos métodos descritos anteriormente, como é possível ver na figura 2-2. Olhando para aquela figura podemos perceber que um carro a esta velocidade possui pertinência 0,4 no conjunto dos rápidos.

O fato de um determinado valor crisp possuir uma pertinência não zero a um conjunto fuzzy, não significa que ele necessariamente possuirá uma pertinência zero em uma conjunto que represente um termo fuzzy conceitualmente oposto àquele primeiro. Por exemplo, o fato de que um carro trafegando a 65 km/h possui uma pertinência não zero no conjunto dos carros rápidos não implicará necessariamente que ele terá pertinência zero no conjunto dos carros lentos.

A pertinência desta velocidade no conjunto dos carros lentos depende apenas da definição deste conjunto. Uma representação possível dos dois conjuntos (lentos e rápidos) pode ser vista na figura 2-2.

Este exemplo evidencia o fato de que um valor associado a uma variável (valor crisp) pode pertencer a dois ou mais conjuntos fuzzy associados ao mesmo conceito linguístico. Isto quer dizer que dois conjuntos fuzzy que representam conceitos opostos (como o conjunto dos carros rápidos com o conjunto dos carros lentos) podem se sobrepor sem nenhum problema.

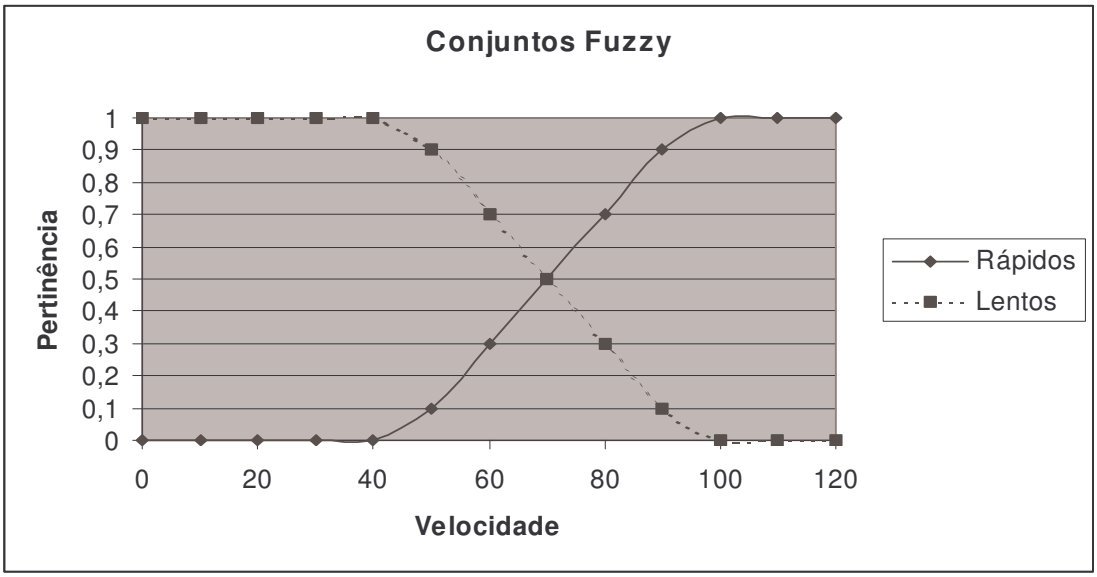


Figura 2-2: Exemplo da definição de dois conjuntos expressando conceitos linguísticos antagônicos para uma mesma variável.

A existência da insterseção é a principal diferença em relação à lógica tradicional. Nesta, se dois termos são conceitualmente opostos, os conjuntos que os definem devem ser disjuntos. Por exemplo, poderíamos definir o conjunto dos carros rápidos como sendo aqueles que trafega acima de 80 km/h, o que faria com que a pertinência de todas as velocidades seja dada por:

$$\mu_{rápido}(vel) = \begin{cases} 0, vel \leq 80 \\ 1, vel > 80 \end{cases}$$

Neste caso, o conceito dos carros lentos seria dado por todos aqueles que não são rápidos, isto é a pertinência no conjunto dos lentos seria dada por:

$$\mu_{lento}(vel) = 1 - \mu_{rápido}(vel)$$

Pode-se verificar que no caso do exemplo mostrado na figura 2-2 isto também é verdade, mas não implica em que, quando a velocidade possui pertinência não zero no conjunto dos carros rápidos, ela necessariamente possua pertinência zero no conjunto dos carros lentos.

É importante ressaltar que o conceito de pertinência fuzzy usado até o momento é totalmente distinto daquele de probabilidade. As incertezas de tipo um, tratadas através de métodos estatísticos e aos quais se aplica o conceito de probabilidade, são aquelas derivadas de comportamento aleatório de sistemas físicos, como flutuações aleatórias de

elétrons em campos magnéticos, flutuações incertas de padrões climáticos, entre outros, enquanto que a incerteza do tipo dois, cuja modelagem pode ser feita através da lógica fuzzy, é aquela que lida como fenômenos decorrentes do raciocínio e cognição humanos (GUPTA *et al.*, 1991). Estes fenômenos são determinísticos, dado o completo conhecimento do processo de raciocínio associado à sua definição.

Pertinência fuzzy é uma incerteza determinística - na lógica fuzzy estamos preocupados com o grau em que algo ocorreu, não com a probabilidade de sua eventual ocorrência. Exemplo: quão alto é uma pessoa, quão rápido está um carro, etc. Os valores da altura e da velocidade são conhecidos a priori, não havendo qualquer tipo de aleatoriedade no processo. No caso do exemplo usado até agora, temos absoluta certeza de que o carro em questão está trafegando a 65 km/h e a sua pertinência de 0,4 no conjunto dos rápidos procura modelar o fato de que ele está "mais ou menos" rápido e não qualquer tipo de incerteza quanto a qualquer erro no processo de medição desta velocidade.

Esta incerteza não se dissipa com o conhecimento do fato, ao contrário do que acontece com a probabilidade. A probabilidade consiste em um conhecimento prévio à ocorrência de um fato que delimita as chances de que ele efetivamente venha a ocorrer. Após a determinação do fato, a probabilidade se extingue, o que não ocorre com o conhecimento fuzzy. Isto quer dizer que a incerteza probabilística se dissipa com as ocorrências, enquanto que a pertinência fuzzy permanece inalterada não importando o número de medições efetuadas.

A pertinência fuzzy descreve uma ambigüidade inerente ao evento, enquanto que a probabilidade descreve as chances de sua eventual ocorrência. Isto é, se um evento ocorre ou não é algo aleatório (probabilístico) enquanto que o grau em que isto ocorre é fuzzy.

Seja, por exemplo, o ato de jogar uma moeda normal, não viciada, para cima, para o qual têm-se uma chance de 50% de obter uma cara. Quando a moeda cai no chão, ou ela o faz com a face da cara para cima ou com a face da coroa para cima. Isto implica em que qualquer incerteza tenha sido dissipada – agora existe 100% de certeza que uma cara ocorreu ou não.

No caso do exemplo do carro usado até agora, a ocorrência da medição precisa da velocidade do veículo não dissipou o fato de que esta (65km/h) faz com que o veículo tenha

pertinência 0,4 (ou 40%) no conjunto dos carros rápidos. Pode-se fazer centenas de medições que este fato não será alterado.

Uma vez compreendido o conceito dos conjuntos fuzzy, precisamos compreender como funcionam os controladores que neles são baseados. Estes controladores usam conjuntos fuzzy associados a regras para realizar o controle e/ou previsão de alguma variável de interesse.

Obviamente, necessitamos definir os conjuntos nos quais os valores reais serão enquadrados. No caso do exemplo da figura 2-2, definimos dois conjuntos no espaço de variáveis (lentos e rápidos), mas podemos definir cinco, doze, ou qualquer outro número que nos seja conveniente. Isto faz com que definamos conjuntos em que um dado valor pode ser enquadrado. Voltando ao exemplo do carro, se definíssemos cinco conjuntos, poderíamos denominá-los de "rapidíssimos", "rápidos", "média velocidade", "lentos", "extremamente lentos". O número de conjuntos nos diz quão precisamente estamos lidando com uma variável. Um exemplo de como estes conjuntos poderiam ser definidos pode ser visto na figura 2-3.

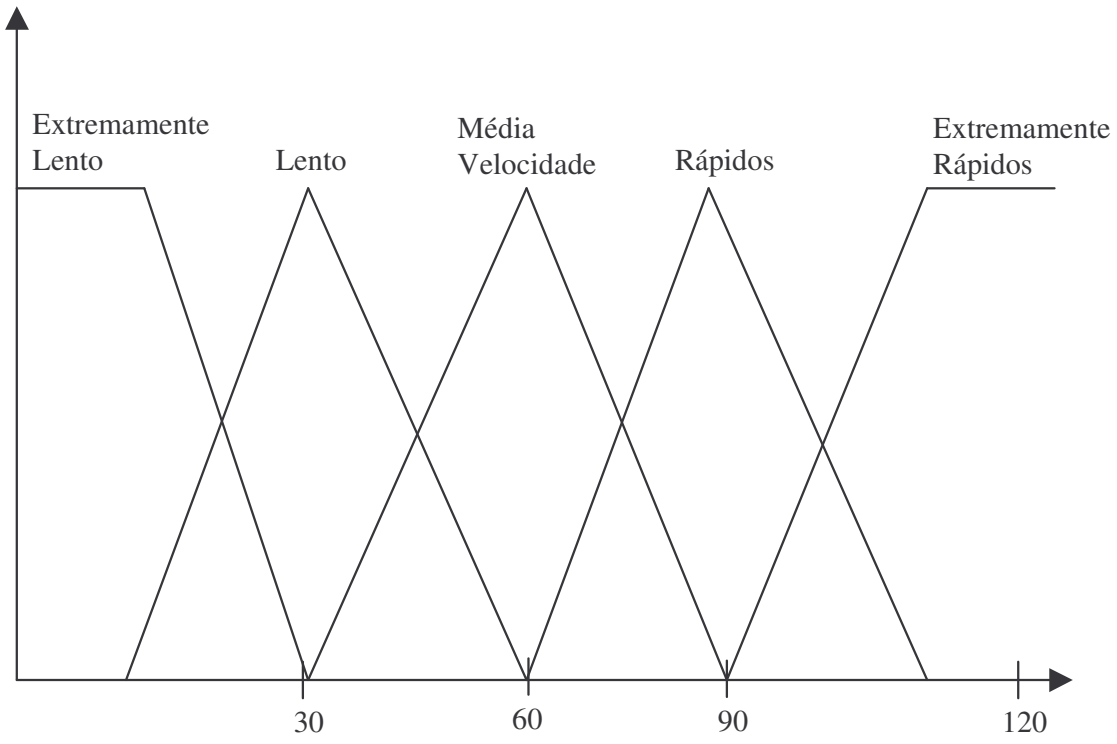


Figura 2-3: Definição de cinco conjuntos fuzzy para a variável velocidade.

Uma vez definidas estas funções, faz-se então um mapeamento das variáveis de entrada em um sistema de controle fuzzy para conjuntos de funções consecutivas, processo este denominado de "fuzzyficação". Para cada valor crisp podemos associar *n* valores fuzzy (um para cada função fuzzy definida). Para todos os valores crisp, vários dos valores fuzzy que lhe são associados podem ser zero.

É improtante ressaltar que nada impede que um sistema de controle possa ter tipos de entradas chaveadas (on/off) junto com entradas analógicas. Tais entradas (on/off) terão sempre um valor verdadeiro igual a ou 1 ou 0.

Estas entradas representam apenas um caso simplificado de uma variável fuzzy o que faz com que o sistema pode negociar com elas sem dificuldade, bastando tratá-las como variáveis fuzzy como todas as outras.

No caso do algoritmo proposto nesta tese, veremos mais à frente que lidaremos não só com variáveis digitais (ou Booleanas), mas também com outro tipo de dado, denominado categórico, que não pode ser expresso de forma numérica e que, por conseguinte, não pode ser tratado usando-se lógica fuzzy. Necessitaremos misturar o tratamento de conjuntos fuzzy com conjuntos tradicionais, além das variáveis Booleanas. Mais detalhes serão vistos no capítulo 6 desta tese.

Assim como na lógica convencional, definimos regras através das quais criamos as associações entre as entradas e saídas. Por exemplo, na lógica digital, quando definimos uma regra do tipo $a \wedge b \rightarrow c$, isto significa que quando *a* E *b* assumirem valores verdadeiros a variável c será verdadeira, caso contrário será falsa. Existem tabelas verdade semelhantes para os operadores OU e NÃO, além de outros operadores que podem ser representados através da combinação destes operadores básicos. As tabelas verdade dos operadores tradicionais são dadas a seguir. Em cada uma delas, o número 1 representa o valor verdade (TRUE) e 0 representa o valor falso (FALSE):

A AND B		
	B=0	B=1
A=0	0	0
A=1	0	1

Tabela 2-1: Tabela Verdade para o operador lógico AND (E)

A OR B		
	B=0	B=1
A=0	0	1
A=1	1	1

Tabela 2-2: Tabela Verdade para o operador lógico OR (OU)

NOT A	
A=0	1
A=1	0

Tabela 2-3: Tabela Verdade para o operador lógico NOT (NÃO)

Quando estamos utilizando a lógica fuzzy, ao definirmos uma regra, o fazemos associando a pertinência das variáveis de entrada em conjuntos determinados à pertinência da variável de saída em um outro conjunto, usando conjuntos fuzzy como definimos previamente e versões fuzzy dos operadores lógicos.

Por exemplo, poderíamos ter uma regra dizendo que quando o carro estiver rápido e a distância para o sinal for pequena, então devemos frear fortemente (em uma representação mais compacta, Rápido(v) AND Pequena(d) $\rightarrow$ Forte(f)).

Precisamos então definir operadores lógicos fuzzy que forneçam, nas condições de contorno, valores lógicos similares aos operadores lógicos tradicionais. Para o NOT, o operador mais simples consiste simplesmente em NOT A = 1 - $\mu_x(A)$, onde $\mu_x(A)$ designa o grau de pertinência do evento *A* no conjunto fuzzy *x*. Já para o caso dos operadores AND e OR, existem vários operadores que satisfazem as condições de contorno, entre os quais podemos destacar os seguintes duplas:

- A AND B = min ($\mu_x(A)$, $\mu_x(B)$)
- A OR B = max ($\mu_x(A)$, $\mu_x(B)$)

e

- A AND B = $\mu_x(A) * \mu_x(B)$
- A OR B = $\mu_x(A) + \mu_x(B) - \mu_x(A) * \mu_x(B)$

Em todas as fórmulas dadas, $\mu_x(A)$ e $\mu_x(B)$ designam respectivamente as pertinências dos eventos A e B no conjunto *x*. Quando o nome do conjunto é significativo, muitas vezes se omite o símbolo μ, designando-se a pertinência pelo nome do conjunto. Por exemplo, ambos os símbolos Rápido(v) e $\mu_{\text{rápido}}(v)$ denotam a pertinência de uma velocidade v no conjunto dos carros rápidos.

Usando qualquer um destes dois operadores podemos combinar pré-condições e determinar a pertinência de um conseqüente. Por exemplo, imagine que temos uma regra que diz que Rápido(v) E Pequena(d) → Forte(f)). Se usamos o primeiro conjunto de operadores (denominado min-max) e a pertinência de Rápido(v) é 0,5 e a de Pequena(d) é 0,4, então podemos determinar que a pertinência do conseqüente Forte(f) é igual a min(0,5 , 0,4)= 0,4.

Sabendo calcular este valor, só é necessário entender como isto será usado no processo de decisão fuzzy. Este processo é baseado no esforço coletivo não só de uma regra como esta que colocamos como exemplo, mas sim de um conjunto de regras (a base de regras). Todas as regras que aplicamos são invocadas, usando as funções consecutivas e valores de pertinência das entradas, de forma a determinar um resultado, que basicamente consiste em uma pertinência da variável de saída a um conjunto específico.

Por exemplo, seja uma base de três regras para o controle da frenagem de um veículo dadas por:

- Rápido(v) E Pequena(d) → Forte(f))
- Lento(v) E Pequena(d) → Média(f)
- Lento(v) E Média(d) → Fraca(f)

O processo de decisão envolvendo lógica fuzzy consiste em primeiro obter os valores de velocidade e distância instantâneos do veículo e depois fuzzyficar estes valores, de acordo com conjuntos previamente definidos (como aqueles da figura 2-2).

Uma vez fuzzyficados estes valores, são aplicados os operadores lógicos e obtemos um valor de pertinência para cada um dos conjuntos que se encontra no conseqüente das regras desta base.

O processo de inferência mais usado é denominado *min-max*, para o qual é adotada a definição de mínimo para o operador AND e máximo para o operador OR. Aplica-se o operador AND sobre cada uma das pertinências do conseqüente. Como o operador AND

calcula a pertinência da regra com base no mínimo das pertinências envolvidas, ele corresponde à parte *min* do nome.

Se houver mais de uma regra com o mesmo conseqüente, escolhe-se a pertinência máxima para este conseqüente como aquela adota para o mesmo, o que corresponde à parte *max* do nome do método. Esta escolha é justificada se considera-se que várias regras para um mesmo conseqüente podem ser interpretadas como uma única regra em que cada um dos conseqüentes está ligado aos outros pelo concetivo lógico OR. Isto é :

$$\begin{cases} A \rightarrow C \\ B \rightarrow C \end{cases} \Rightarrow A \vee B \rightarrow C$$

Uma vez aplicado o processo de inferência, o resultado de um controlador fuzzy consiste em uma variável de saída para a qual foram definidos vários conjuntos fuzzy, a cada um dos quais foi associada uma pertinência. Entretanto, normalmente o interesse de um usuário de um controlador fuzzy não é nas pertinências aos conjuntos, mas sim em um valor final crisp da variável de saída que possa ser utilizado em um controlador.

Por conseguinte, é necessário ainda uma função que calcule, a partir dos graus de pertinência desta variável a cada um dos conjuntos fuzzy (calculado através das regras), o valor real daquela variável, processo este que é o oposto do processo de fuzzyficação, e por isto, é denominado *defuzzyficação*.

O método de defuzzyficação mais simples é denominado média dos máximos, no qual se calcula a média dos máximos de cada um dos conjuntos fuzzy da variável de saída ponderada pelas pertinências obtidas através do sistema de inferência. Matematicamente:

$$saída_{\text{var}} = \frac{\sum_{i \in conjuntos_{\text{var}}} \mu_i * \max(i)}{\sum_{i \in conjuntos_{\text{var}}} \mu_i}$$

Assim, obtem-se um valor de saída para a variável de interesse que pode ser usado em um controlador, um previsor ou qualquer outra aplicação em que se deseje aplicar a lógica fuzzy.

2.2 Algoritmos evolucionários

Os algoritmos evolucionários são inspirados na teoria da evolução das espécies de Charles Darwin. Na década de 1850 Charles Darwin fez uma longa viagem no navio HMS Beagle, percorrendo uma grande distância náutica e visitando vários países do mundo, inclusive da América do Sul.

Darwin era um observador da natureza e sua habilidade para observação permitiu que ele percebesse vários fatos interessantes. Os principais deles consistiam em que animais da mesma espécie eram ligeiramente diferentes que outros animais na mesma espécie em ecossistemas distintos, sendo que cada grupo era mais adaptado às necessidades e oportunidades oferecidas pelo seu ecossistema específico.

Estes fatos, grosseiramente simplificados neste resumo, levaram Darwin a concluir que havia um processo evolutivo associado ao desenvolvimento das espécies no mundo. Esta teoria levou o nome de "Teoria da Evolução das Espécies".

A teoria da evolução diz que na natureza todos os indivíduos dentro de um ecossistema competem entre si por recursos limitados, tais como comida e água. Aqueles dentre os indivíduos (animais, vegetais, insetos, etc) de uma mesma espécie que não obtêm êxito tendem a ter uma prole menor e esta descendência reduzida faz com que a probabilidade de ter seus genes propagados ao longo de sucessivas gerações seja menor.

A combinação entre os genes dos indivíduos que sobrevivem pode produzir um novo indivíduo muito melhor adaptado às características de seu meio ambiente ao combinar características possivelmente positivas de cada um dos reprodutores.

Um pouco mais adiante, no início do século XX, Mendel compreendeu que este processo de transmissão de características positivas estava associado a uma unidade básica de transmissão de informação, o gene.

Hoje, após as descobertas da estrutura do DNA por Francis e Crick e do papel deste como unidade básica de armazenamento e transmissão de informação genética, sabemos como este processo funciona em nível molecular.

Basicamente, todo indivíduo, seja ele animal, vegetal ou mesmo organismos inferiores como vírus e bactérias, possui um conjunto de um ou mais cromossomos e a este conjunto completo denominamos genoma.

Um conjunto específico de genes no genoma é chamado de genótipo. O genótipo é a base do fenótipo, que é a expressão das características físicas e mentais codificadas pelos genes e modificadas pelo ambiente, tais como cor dos olhos, inteligência, etc.

Nos organismos que utilizam a reprodução sexuada, como os humanos e as moscas, cada progenitor fornece um pedaço de material genético chamado gametas. Estas gametas são resultado de um processo denominado crossing-over, ilustrado na figura 2-4 que permite que os filhos herdem características de seus pais mas não sejam exatamente iguais a estes. Além disto, mutações causadas por fatores aleatórios tais como presença de radiação ambiente ou erro nos mecanismos de replicação do DNA podem causar pequenas mudanças nos genes dos indivíduos.

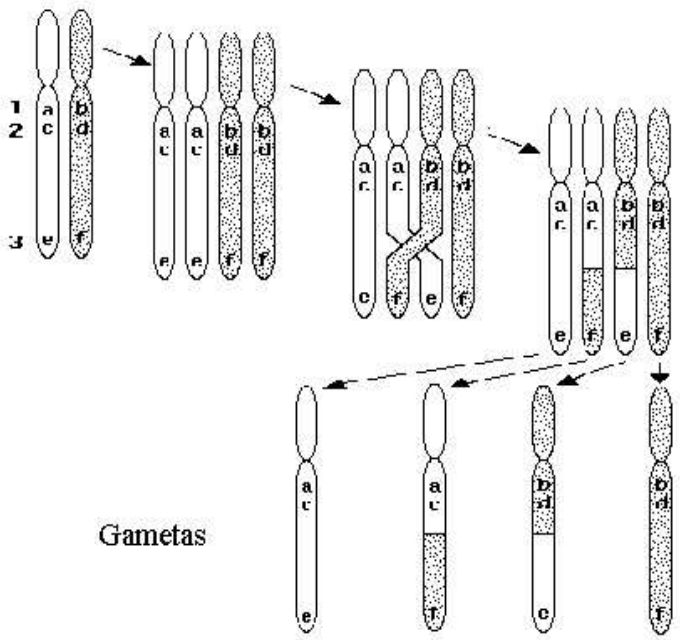


Figura 2-4: Diagrama do processo de crossing-over ocorrido durante a formação de gametas para a reprodução sexuada.

Os Algoritmos Evolucionários (EA) são modelos computacionais dos processos naturais de evolução usados como uma ferramenta para resolver problemas. Apesar de haver uma grande variedade de modelos computacionais propostos, todos eles têm em

comum o conceito de buscar uma solução para um problema através da simulação da evolução das espécies usando mecanismos de seleção, mutação e reprodução similares àqueles encontrados na natureza. Todos estes processos, assim como os seus análogos naturais, dependem do desempenho dos indivíduos de uma espécie dentro do "ambiente" que a rodeia.

Os EA funcionam mantendo uma população de estruturas (indivíduos), cada qual propondo uma determinada solução para um problema. Estas estruturas evoluem de forma semelhante aquela que é observada na natureza.

Para que isto aconteça estas estruturas são submetidas a forças de seleção que pressionam pela sobrevivência das estruturas que melhor resolvem o problema (as "mais aptas") ao mesmo tempo em que são aplicados os chamados operadores genéticos, como recombinação e mutação, entre outros.

Os algoritmos evolucionários são usualmente aplicados para resolver um problema específico de caráter NP-completo (MITCHELL, 1996). Sua relação como este problema é dada através da função de avaliação, a qual representa computacionalmente o problema, na forma de uma função a ser otimizada.

Esta função é aplicada a cada indivíduo da população de estruturas mantida pelo EA e estes recebem uma avaliação que é uma quantificação numérica da sua qualidade como solução do problema.

Esta avaliação é usada como a base do processo de seleção artificial mantido pelos EA. Quanto mais alta a avaliação do indivíduo, "mais apto" ele é considerado e por conseguinte, maiores devem ser suas chances de sobrevivência e reprodução. Assim como na natureza, é provável que a prole de um indivíduo compartilhe de suas boas características e por conseguinte, é desejável que indivíduos mais aptos reproduzam mais, de forma que seja gerada uma população de descendentes com uma avaliação média mais alta.

Os operadores genéticos utilizados nos EA consistem em modelos computacionais de fenômenos vistos na natureza, como a reprodução sexuada, a mutação genética e quaisquer outros que a imaginação dos programadores consiga reproduzir. Existem vários modelos distintos de operadores genéticos, alguns dos quais serão discutidos de forma mais detalhada no decorrer desta tese.

O comportamento padrão dos algoritmos evolucionários pode ser resumidos, sem maiores detalhes pelo seguinte pseudo-código, descrito em linguagem natural para maior generalidade, permitindo então que a implementação seja feita em qualquer linguagem de programação.

```
t:=0 // Inicialize o contador de tempo
Inicializa_População P(0) // Inicialize a população aletoriamente
Enquanto não terminar faça // término:por tempo, por avaliação, etc.
      Avalie_População P(t) //Avalie a população neste instante
      P':=Selecione_Pais P(t)  // sub-população que gerará nova geração
      P'=Recombinação_e_mutação P' //Aplique os operadores genéticos
      P(t+1)=Selecione_sobreviventesP(t),P' //sobreviventes da geração
      t:=t+1 //Incremente o contador de tempo
Fim enquanto
```

Neste algoritmo pode-se perceber que o comportamento básico dos algoritmos evolucionários consiste em buscar dentro da atual população aquelas soluções que possuem as melhores características e tentar combiná-las de forma a gerar soluções ainda melhores, repetindo este processo até que tenha se passado tempo suficiente ou que tenhamos obtido uma solução satisfatória para nosso problema.

Este processo é similar ao processo evolutivo que ocorre na natureza e assim como nesta, os EA não são um processo orientado à otimização. Isto pode parecer paradoxal mas não existem mecanismos na evolução que asseguram o aprimoramento contínuo dos indivíduos. Esta melhora ocorre naturalmente através da seleção natural.

Um ponto fundamental em relação aos algoritmos evolucionários é que, como se pode perceber claramente dentro do algoritmo proposto acima, os algoritmos evolucionários são dependentes de fatores estocásticos (probabilísticos), tanto na fase de inicialização da população quanto na fase de evolução (durante a seleção dos pais, principalmente).

Pode-se concluir então que, na prática, os EA são uma heurística e os seus resultados provavelmetne não serão iguais em todas as rodadas, fazendo com que seus resultados, de uma forma geral, não sejam reprodutíveis. O que se faz, em geral, é adotar a média de um determinado número de melhores soluções como a solução proposta pelo EA

ou então escolher a solução com avaliação mais alta dentre todas aquelas oferecidas por um EA para ser a resposta oferecida por esta heurística.

Aqueles que desejem aplicar EA a algum problema real devem estar conscientes da limitação da reprodutibilidade de seus resultados. Ademais, posto que fatores estocásticos têm uma influência direta sobre o desenvolvimento de uma solução, execuções distintas podem obter resultados radicalmente diferentes, fazendo com que a variância e o desvio-padrão associado a múltiplas soluções sejam altos.

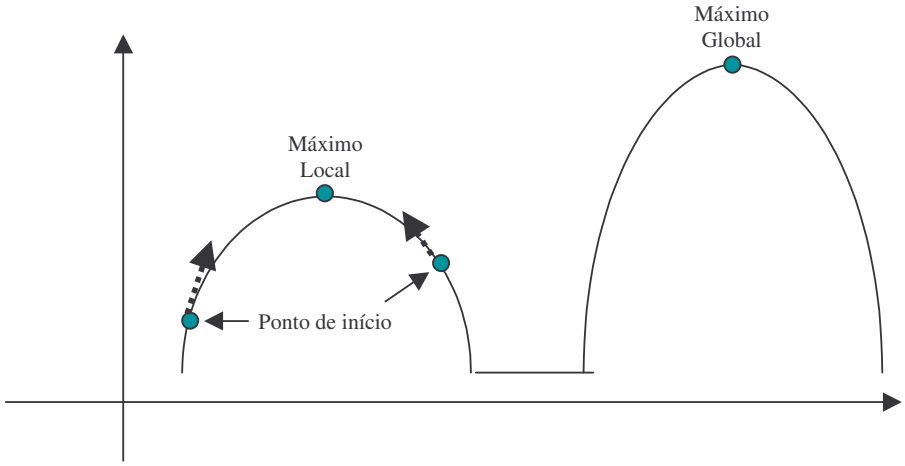


Figura 2-5: Exemplo de aplicação de um algoritmo de hill climbing a uma função multimodal F. Caso a inicialização não se dê próxima ao máximo local, o algoritmo seguirá o gradiente da função até chegar ao máximo local, o qual ∇F=0, implicando no fim do algoritmo, como podemos ver nos dois casos exemlpificados na figura. Ao se iniciar em qualquer um destes pontos, o algoritmo de hill climbing parará no máximo local apontado na figura, sendo incapaz de reconhecer a existência de outro máximo, já que dispõe apenas de informações topográficas locais sobre a função F.

Algoritmos genéticos (GA) são um ramo dos algoritmos evolucionários e como tal podem ser definidos como uma técnica de busca baseada numa metáfora do processo biológico de evolução natural.

Os algoritmos genéticos são técnicas heurísticas de otimização global. A questão da otimização global opõe os GAs aos métodos como hill climbing, que seguem a derivada de uma função de forma a encontrar o máximo de uma função, ficando facilmente retidos em máximos locais, como podemos ver na figura 2-5.

Nos algoritmos genéticos populações de indivíduos são criados e submetidos aos operadores genéticos, que usualmente consistem em seleção, crossover e mutação. Estes operadores utilizam uma caracterização da qualidade de cada indivíduo como solução do

problema em questão chamada de avaliação deste indivíduo e vão gerar um processo de evolução natural destes indivíduos, que eventualmente gerará um indivíduo que caracterizará uma boa solução (talvez até a melhor possível) para o nosso problema.

Definindo de outra maneira, podemos dizer que algoritmos genéticos são algoritmos de busca baseados nos mecanismos de seleção natural e genética. Eles combinam a sobrevivência entre os melhores indivíduos com uma forma estruturada de troca de informação genética entre dois indivíduos para formar uma estrutura heurística de busca (MITCHELL, 1996). Esta informação genética está armazenada nos esquemas contidos dentro dos cromossomos.

Um esquema consiste em um template descrevendo um subconjunto dentre o conjunto de todos os indivíduos possíveis. O esquema descreve similaridades entre os indivíduos que pertencem a este subconjunto, ou seja, descreve quais posições dos seus genomas são idênticas.

O alfabeto de esquemas consiste no alfabeto de símbolos utilizados na representação adotada mais o símbolo *, que significa "não-importa" (don't care ou wildcard), isto é, que os indivíduos que correspondem àquele esquema diferem exatamente nas posições onde este símbolo é encontrado.

Formalmente, pode-se definir um esquema como sendo uma string $s=\{s_1\ s_2\ ...\ s_N\}$, de comprimento n, cujas posições pertence ao conjunto Γ (alfabeto usado) + {*} (símbolo de wildcard). Cada posição da string dada por $s_k \neq$ '*' é chamada de especificação, enquanto que o símbolo * especifica um wildcard.

O teorema dos esquemas (HOLLAND, 1975) diz que um esquema ocorrendo nos cromossomos com avaliação superior à média tende a ocorrer mais frequentemente (com a freqüência crescendo de forma exponencial) nas próximas gerações e aqueles esquemas ocorrendo em cromossomos com avaliações abaixo da média tendem a desaparecer.

2.3 Redes de regulação genética

A maioria das células de um mesmo organismo tem o mesmo DNA (algumas diferindo por fatores como rearranjos e amplificação), entretanto as células são diferentes entre si. É impossível confundir uma célula do fígado com um neurônio, ou mesmo uma célula muscular com uma célula de tecido adiposo, não só pela diferença óbvia de função e aparência mas também pelos diferentes genes expressos em cada um destes tipos de células.

Esta expressão diferenciada dos genes é fundamental para a formação de padrões e durante o desenvolvimento de um organismo multicelular e ocorre por muitos fatores, incluindo presença de elementos sinalizadores no ambiente celular (como hormônios), sinalização de célula para célula e regulação transcripcional.

Esta regulação ocorre pela ação combinatorial de fatores de transcrição (produtos de outros genes) nos elementos situados "próximos" ao ponto de início da transcrição dos genes. Isto quer dizer que os produtos da expressão de um gene afetam outros genes que são seus vizinhos promovendo ou inibindo sua expressão.

O conceito de proximidade aqui não necessariamente corresponde ao físico, mas sim em termos de adjacências em um grafo que represente o esquema de regulação de cada gene. Isto é, genes expressos em pontos que sejam mutuamente distantes dentro do genoma podem pertencer a uma cadeia de regulação e se afetar mutuamente.

Este grafo é formado tendo como nós cada um dos genes e as arestas determinadas pelo fato de um dos genes ser afetado/controlado pelo nível de expressão do outro. Com este esquema cria-se um grafo fortemente interconectado que representa a rede de regulação genética, rede esta que pode ser definida como sendo o somatório de todas as interconexões existentes no processo regulatório. Este conceito pode ser compreendido mais facilmente observando-se a figura 2-6. O conceito de redes de regulação será discutido de forma mais detalhada mais à frente nesta seção.

Este controle transcripcional é essencial para o estabelecimento da expressão diferenciada, formação de padrões e desenvolvimento do organismo como um todo. Cada tipo de célula é diferente devido ao fato dos diferentes genes que nela estão expressos. Isto significa que em muitos casos duas células são muito diferentes apesar do fato de compartilharem a mesma informação genética.

Pode-se concluir então que a causa da diferença entre tipos celulares não reside no genoma, mas no conjunto de genes que está sendo expresso em casa célula. Por exemplo, genes que codificam para certas enzimas especiais que são necessárias apenas em células hepáticas não estarão ativos em neurônios, da mesma maneira que os genes que codificam para neurotransmissores não estarão expressos em células hepáticas (ALBERTS *et al.*, 2002).

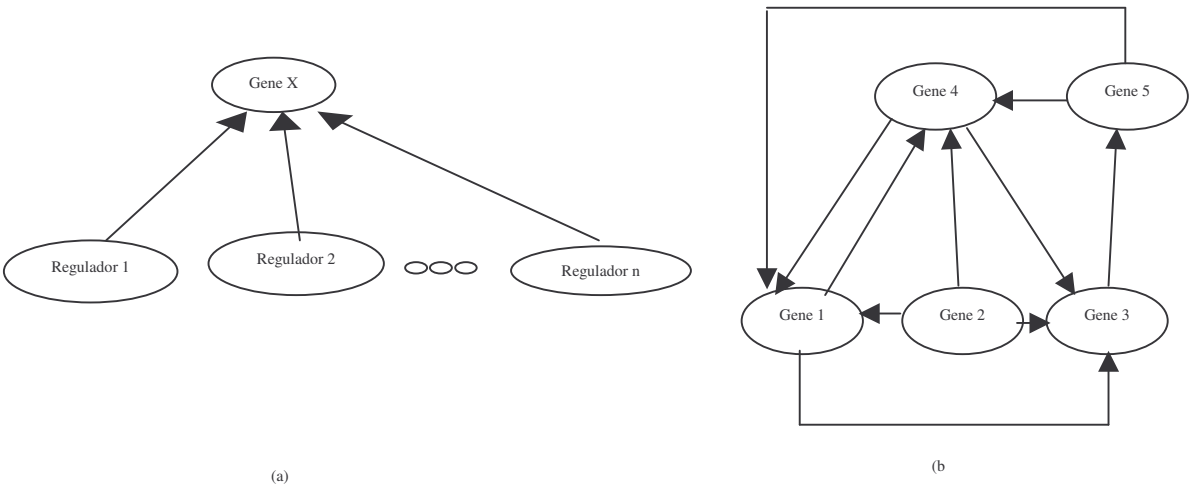


Figura 2-6: Modelo do grafo formado pelas redes de regulação celular. Em (a) nós vemos um gene sendo regulado por vários outros genes reguladores. Entretanto, esta diferença entre regulados e reguladores não existe. A realidade é como mostrada na figura (b) em que vemos um modelo de rede formado por cinco genes que assumem o papel de regulados e reguladores dependendo de qual processo se está considerando.

Já foi estabelecido, conforme mencionado anteriormente nesta tese, que as redes de regulação são altamente conectadas. Em metazoa superiores, estima-se que cada gene ou proteína interage com um número de genes que varia de quatro a oito (ARNONE *et al.*, 1997), além de estar envolvido em cerca de dez funções biológicas. Isto desqualifica a maioria dos métodos computacionais que procuram limitar as interações a um número constante que torne o problema computacionalmente tratável, como por exemplo em (AKUTSU *et al.*, 1999).

A idéia de limitar o número de entradas é interessante em termos computacionais. O trabalho citado demonstra que número de funções Booleanas existentes em uma rede cujo número de interações é limitado por um número K é da ordem de $\Omega((2^{2^K} * n^K)^n)$, onde n consiste no número de genes no conjunto sendo avaliado. Isto é, se permitirmos que o número K cresça de forma ilimitada, o número de funções será alto demais.

Entretanto, a idéia de limitar o número de interações entre genes está em total desacordo com a biologia por trás do problema. Assim, qualquer abordagem que siga por este caminho deve ser desconsiderada.

Como será ressaltado também nas aplicações de classificação demonstradas no capítulo 6, é importante compreender a ciência geradora dos dados que estão sendo usados, de forma que os resultados obtidos tenham uma máxima aplicabilidade.

Figura 2-7: O dogma central da biologia

Sem um estudo cuidadoso da biologia de forma que o algoritmo adotado respeite as restrições e características do problema em questão, as soluções obtidas terão pouca relevância para os usuários finais da informação. Isto posto, faremos seguir uma breve exposição dos princípios nos quais se baseiam a área da biologia à qual pertence a aplicação descrita neste capítulo.

O dogma central da biologia, mostrado na figura 2-7, estabelece que somente uma pequena fração do DNA será transcrita em mRNA e este será traduzido posteriomente em proteína.

À primeira vista pode parecer contraintuitivo que outro composto, como o RNA, seja utilizado como um intermediário entre o DNA e a codificação de proteínas. Afinal, a adição de mais um passo acrescenta mais um ponto de erro e cria a necessidade de um máquinário biológico complexo dentro da célula para realizar o processo de transcrição.

Entretanto, a evolução "optou" pelas células que usassem este passo intermediário por alguns motivos principais:

- Nas células eucarióticas, o DNA pode permanecer protegido do ambiente cáustico do citoplasma celular.
- A informação celular pode ser amplificada através da criação de múltiplas cópias de RNA a partir de uma única cópia de DNA.
- A regulação da expressão gênica pode ser afetada através da criação de controles específicos do caminho entre o DNA e as proteínas. Quanto mais elementos houver neste caminho, mais oportunidades de controle existem para aplicação nas mais diversas circunstâncias.

O dogma central descrito acima não é totalmente verdadeiro. Em alguns experimentos pôde-se verificar que a quantidade de proteína não é perfeitamente correlacionada com a quantidade de seu mRNA codificador (na verdade, a correlação se aproxima mais de 0,5) (ALBERTS *et al.*, 2002). Isto se deve a outros fatores não embutidos no dogma central, como o controle da degradação das moléculas de mRNA e entrada de substâncias vindas do ambiente extra-celular, entre outros

Conseqüentemente, deve-se entender que a expressão gênica e a conseqüente síntese de proteínas é um processo complexo regulado em diversos estágios. Além da regulação da transcrição de DNA, que é a forma mais estudada de controle, a expressão de um gene pode ser controlada durante o processamento de RNA e o transporte do núcleo para a matriz celular (somente nas células eucarióticas, tendo em vista que as procarióticas não possuem núcleo), na tradução do RNA e até mesmo através da modificação das proteínas após o processo de tradução.

Ademais, deve-se entender que as proteínas não se degradam ao mesmo tempo, o que implica que sua degradação, bem como a de produtos intermediários de RNA, também pode ser regulada dentro da célula.

Tendo em vista que estas proteínas que realizam esta regulação são codificadas por outros genes da mesma ou de outra célula, surge um verdadeiro sistema regulatório genético estruturado através de uma rede de interações regulatórias entre DNA, RNA, proteínas e pequenas moléculas (SMOLEN *et al.*, 2000).

Todos estes fatores levam à conclusão de que o padrão de expressão gênica resulta de uma interação de caminhos regulatórios individuais. Nestas redes de sinalização celular altamente conectadas, as funções de um gene dependem de seu contexto celular e possivelmente do comportamento de todos os outros genes à sua volta. Estas redes podem ser modeladas por grafos, conforme discutido acima e visto na figura 5-1, para uma melhor compreensão.

Logo, o objetivo de entender esta rede como um todo é essencial. Por muitos anos este tipo de trabalho não foi realizado, apesar de a comunidade científica estar plenamente ciente de sua importância devido à grande complexidade associada a compreenderem-se sistemas dinâmicos com grande número de variáveis e a dificuldade de se conseguir dados suficientes para o tratamento deste problema. Esta última barreira se quebrou, ao menos parcialmente, com o surgimento de novas ferramentas para o estudo da genômica, como os microarrays de DNA, discutidos a seguir.

2.4 Microarrays de DNA

A criação de novas ferramentas causou uma revolução na genômica e na maneira como os cientistas entendem o processo de expressão gênica, dado o grande fluxo de informação que tem sido gerado. Hoje é óbvio que o entendimento corrente dos fenômenos biológicos, como a expressão diferenciada, não será dificultado pela falta de dados, mas sim pela dificuldade de analisar e interpretar os dados disponíveis de forma adequada. Uma destas tecnologias que teve um grande impacto nas ciências biológicas foi o desenvolvimento da tecnologia de microarrays de DNA.

Os genes transcritos em mRNA são denominados de genes expressos e a sua análise é denominada análise de expressão gênica (gene expression analysis). Cada gene é expresso de forma diferenciada dentro da célula e a quantidade de cada tipo de mRNA determina o estado atual da célula. Se estas quantidades forem medidas continuamente, pode-se ter uma melhor idéia do funcionamento das redes de regulação como um todo.

A idéia dos microarrays é monitorar as interações de um conjunto de fragmentos de DNA com um conjunto pré-determinado de *probes*, isto é, seqüências pré-determinadas, que ficam presas em uma placa de vidro. Estas placas são chamadas de chips de DNA e permitem que milhares de hibridizações sejam realizadas ao mesmo tempo. O princípio básico por trás do funcionamento dos microarrays é mostrado na figura 2-8.

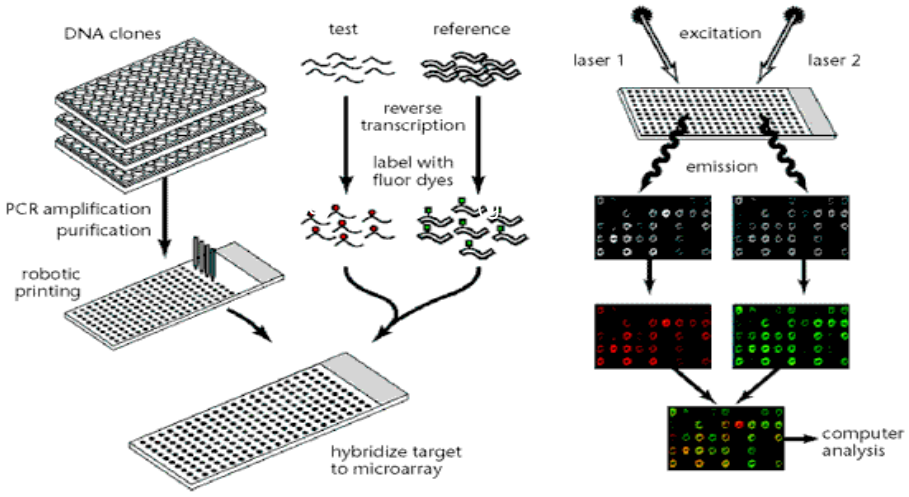


Figura 2-8: Diagrama de operação dos microarrays de tecnologia Synteni/Stanford. Este tipo de chip mede o nível de expressão de uma amostra relativo a uma amostra de base. Cada uma das amostras é pigmentada com uma cor diferente e lasers de duas freqüências analisam os níveis relativos de expressão, após uma mistura de duas amostras ser hibridizada com os probes do slide. Figura originalmente publicada em (DUGGAN et al., 1999)

O processo consiste basicamente em imprimir uma lâmina de vidro com arrays de DNA usando impressão de pequenos volumes (da ordem de nanolitros ou picolitros) de uma amostra pré-seqüenciada (probe) em uma superfície de vidro pré-preparada (Baldi, 2002). Cada ponto do microarray consiste em um gene completo ou em uma seqüência pré-determinada, previamente condificada (fig. 2-9).

O DNA das amostras a serem seqüenciadas é ligado a um pigmento específico e colocado nas placas de vidro e as seqüências que são complementares se ligarão. A presença de DNA ligado é depois detectada através da excitação por um laser cuja freqüência de excitação é muito próxima à freqüência máxima do pigmento.

Existem várias diferentes tecnologias de microarrays de DNA. Elas diferem em como as seqüências de DNA são impressas no chip (fotolitografia, spotting usando robôs, etc.), no comprimento das seqüências de DNA (seqüências completas ou apenas

fragmentos), mas todas elas têm em comum a característica de permitir que sejam medidos os padrões de expressão de alguns genomas quase completos.

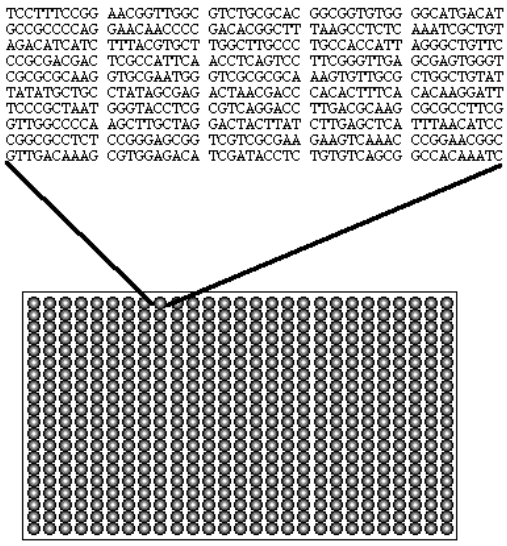


Figura 2-9: Cada ponto de um microarray consiste em uma seqüência genética de interesse (possivelmente, um gene completo)

Ao permitir que os padrões de expressão de milhares de genes sejam medidos simultaneamente, os microarrays fornecem uma visão abrangente do estado corrente da célula em questão. Isto decorre do fato já discutido acima de que grande parte das diferenças de estado e tipo de células está fortemente correlacionada com mudanças do nível de mRNA de certos genes.

Assim, a análise do padrão de expressão de vários genes (alguns podendo ser previamente desconhecidos) pode, especialmente se associada com informações relativas aos caminhos (*pathways*) de regulação genética, fornecer pistas sobre a função de vários genes e a forma como eles se relacionam.

Apesar do custo deste tipo de experimento ter caído de forma exponencial nos últimos anos, ainda não é prático, especialmente em laboratórios de orçamento restrito, propor que os biólogos realizem centenas de experimentos de microarrays para desenvolver um modelo biológico.

Esta impossibilidade financeira de realizar diversos experimentos pode se tornar o grande calcanhar de Aquiles dos experimentos de microarrays, dado o problema do ruído experimental e falta de verificação estatística dos resultados.

COOPER *et al.* (2003) afirma que o grande benefício dos microarrays (sua habilidade de medir vários genes ao mesmo tempo) pode ser o seu grande problema, devido ao fato de que é possível observar certas características nos dados (no artigo, ciclos são citados especificamente) que derivam exclusivamente de variações estatísticas resultantes de ruído branco e/ou variações experimentais e que não existiriam realmente no ambiente celular.

O artigo afirma que seria necessário realizar um grande número de experimentos e comparar os resultados dos mesmos com resultados obtidos de experimentos randomizados, de forma a conseguir dados que sejam estatísticamente significativos. Entretanto, conforme mencionamos anteriormente, devido a falta de recursos financeiros, isto pode ser praticamente impossível.

É importante ressaltar que os mesmos autores já haviam publicado anteriormente um artigo (SHEDDEN *et al.*, 2002) em que discutiam a validade de resultados baseado na análise de experimentos com microarrays, mais especificamente a questão da descoberta de comportamento cíclico baseado nestes dados.

Os autores comentam que o grande problema do experimento específico que eles criticam (e que pode ser generalizado para vários outros estudos) é a relativa escassez dos dados, escassez esta devida ao grande intervalo de tempo entre medições e a falta de repetitibilidade dos experimentos.

Os autores do estudo original (CHO *et al.*, 2001), responderam estas afirmações de forma veemente, defendendo a metodologia de seu estudo, mas isto é de pouca relevância no contexto desta tese. Experimentos de microarrays já se mostraram úteis na descoberta de informação biológica e esta discussão entre ambos os autores é de cunho acadêmico entre pesquisadores cujos modelos são igualmente válidos, mas sob condições distintas.

Esta discussão suscita uma preocupação importante, que consiste em uma obrigação de todo cientista sério de levar em consideração os resultados de experimentos isolados (ou mesmo repetidos apenas uma ou duas vezes) de microarrays com cautela, devido a potenciais erros de medição e da falta de significância estatística inerente a experimentos sujeitos a fatores estocásticos, como é o caso das medições de microarrays, que foram repetidas poucas vezes.

Não se deve desmerecer a importância desta relevante fonte de dados, que tem se mostrado útil na descoberta de vários novos aspectos da regulação celular. O que é necessário, neste momento, é maximizar o valor dos conjuntos de dados que existem, que, por definição, são escassos em número, e direcionar as pesquisas de forma que o pequeno número de experimentos disponíveis possa nos fornecer o resultado máximo em termos de informação sobre a regulação celular.

É difícil quantificar a qualidade de resultados obtidos a partir destes dados. Cada conjunto de dados pode "esconder" uma quantidade diferente de informação que não se conhece a priori, mas que pode ser descoberta por alguma técnica computacional.

Assim sendo, seria muito interessante que se desenvolvesse uma técnica que criasse hipóteses a partir de poucos experimentos, de forma a guiar a pesquisa futura, diminuindo os custos associados a novas descobertas biológicas.

O algoritmo descrito nesta tese, dada a sua capacidade de lidar com dados esparsos, se mostra um candidato adequado para aplicação em tal área. Assim, decidiu-se usá-lo para buscar candidatos a reguladores que sejam promissores, diminuindo assim o número de experimentos necessários para determinação de uma rede de regulação genética.

Capítulo 3 – Algoritmo Booleano Proposto

Como passo preliminar do desenvolvimento do trabalho desta tese, foram estudadas redes Booleanas para modelar redes de regulação genética. Elas são muito simples, mas exibem certos comportamentos análogos ao desenvolvimento dos organismos reais, o que faz com que seja possível utilizá-las para modelar certos fenômenos biológicos, como vemos na tabela 3-1 a seguir.

Fenômeno Biológico	*Análogo em Redes Booleanas*
Tipos celulares (diferentes tipos de células presentes em um organismo)	Atratores no espaço de estados (pontos e/ou ciclos para os quais há convergência das trajetórias da rede e que são estáveis)
Diferenciação celular (transição entre tipos celulares, devida a influências externas ou regulação genética)	Transições entre atratores (a modificação das condições de uma rede faz com que passe a haver uma convergência para outro atrator no espaço de soluções)
Estabilidade de padrões (estabilidade das células mesmo quando da variação de seu ambiente, dentro de certos limites.	Bacias de Atração (espaço no qual todos os pontos convergem para o mesmo atrator)

Tabela 3-1: Correspondência entre fenômenos biológicos e termos associados a redes Booleanas

Tendo em vista as similares citadas, podemos defender o uso de um modelo tão simples quanto uma rede Booleana, pois ela pode nos fornecer uma visão qualitativa de aspectos importantes do complexo processo de desenvolvimento dos organismos.

No caso específico de nosso trabalho, redes Booleanas serviram como um passo intermediário, de preparação para o passo de modelagem através do uso de lógica fuzzy, que nos permitiu modelar valores contínuos no intervalo [0,1] e não ficar restritos a apenas dois estados.

Entretanto, como será discutido a seguir, o modelo Booleano não deve ser descartado totalmente como sendo "primitivo" ou "incapaz de modelar" a complexidade das redes de regulação genética. SZALLASI *et al.* (1998) e LIANG *et al.* (1998) demonstram de forma inequívoca que esta desqualificação deste modelo seria no mínimo injusta e precipitada, tendo em vista os bons resultados (em termos qualitativos) que se pode obter utilizando-as.

3.1 Redes Booleanas

Redes Booleanas foram introduzidas por Kauffman nos anos 60 (KAUFFMAN, 1969) como uma abstração de redes de regulação genética nas quais cada gene é modelado como estando em um dentre dois estados (ligado, ON ou desligado, OFF) e o estado de cada gene no próximo instante de tempo é determinado por uma função Booleana de suas entradas no instante de tempo corrente. (D'HAESELEER, 1993).

Uma rede Booleana, denotada G(V,F) é um conjunto $V=\{v_1,v_2,...,v_n\}$ de nós representando genes e uma lista de funções Booleanas $F=\{f_1,f_2,...,f_n\}$, onde uma função Booleana $f_i(v_{i1}, v_{i2}, ..., v_{ik})$, onde k é o número de nós que regulam o nó i. Esta função, designada para cada nó v_i, recebe entradas dos nós $v_{i1}, v_{i2}, ..., v_{ik}$. Para um determinado subconjunto $U \subseteq V$, um padrão de expressão de U é uma função ψ de U para {0,1}

Um padrão de expressão também é chamado de estado da rede Booleana e representa o estado de todos os nós (genes) onde cada nó pode assumir o valor 0 (não

expresso) ou 1 (expresso) como seu valor em cada instante. O padrão de expressão no instante t+1 é determinado pelas aplicação síncrona das funções Booleanas pertencentes a F ao padrão de expressão no instante t (por exemplo, v_i (t+1) = $f_i(v_{i1}(t), v_{i2}$ (t), ..., $v_{ik}(t))$ (AKUTSU *et al.*, 1999).

Esta sincronicidade simplifica a análise de redes Booleanas, mas faz com que certos tipos de padrões complexos exibidos na natureza não possam ser modelados com total precisão, como será discutido mais na próxima seção.

3.2 Modelando Redes de Regulação Genética com Redes Booleanas

Neste capítulo será descrito um algoritmo de modelagem de redes de regulação genética usando-se redes Booleanas. Este é um modelo reconhecidamente simplificado dado que, entre outras coisas, os efeitos de fatores estocásticos estão ausentes do modelo. DELLAERT (1995) aponta as seguintes implicações para o uso de redes Booleanas:

- Diferenças entre elementos regulatórios são ignorados, havendo apenas um único tipo de elemento genético.
- Todos os elementos genéticos influenciam os outros de forma similar, de modo que as diferenças entre estratégias diferentes de regulação em células biológicas também são ignoradas.
- Elementos genéticos são modelados como estando ativos ou inativos, o que significa que conceitos tais como concentrações e respostas graduais existentes nas redes biológicas são perdidos. Existem vários processos biológicos que respondem seguindo funções sigmoidais ou S-Shaped (fig. 3-1a), o que lhes confere um gradualismo maior do que pode ser obtido utilizando-se um função Booleana, no formato de degrau (fig. 3-1b).

- A transição de todos os elementos de uma rede Booleana ocorre sincronamente, isto é, todos os elementos são atualizados simultaneamente. Apesar de haver alguns processos naturais cíclicos (controlados por uma espécie de "relógio biológico"), grande parte dos processos celulares são assíncronos, ocorrendo quando as pré-condições que lhes são necessárias são satisfeitas (acúmulo de uma proteína, ausência de um inibidor, etc.). Este assincronismo cria uma complexidade maior no comportamento celular que não pode ser reproduzido facilmente por um processo síncrono.

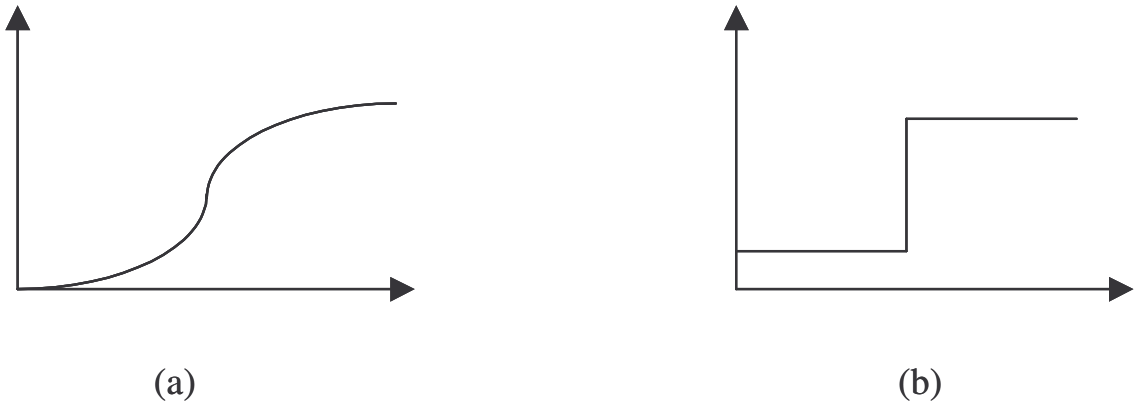


Figura 3-1: (a) Uma função sigmóide típica. A resposta é gradual, não havendo pontos de derivada indefinida. (b) Uma função degrau típica. Quando determinadas condições são satisfeitas, há uma transição imediata dos valores de saída. Neste ponto a derivada da função é infinita.

Apesar destas limitações, existem várias características da regulação celular que se assemelham a um comportamento Booleano. Há exemplos de expressão de DNA nos quais a inibição se dá através da ligação de um elemento (que ou está ligado ou não, configurando um comportamento Booleano) a um operon (ALBERTS *et al*, 2002) e muitos outros elementos são regulados através de um processo signmoidal, que pode ser aproximado de forma razoável como uma função degrau, que é essencialmente on-off, ou booleana.

Existem ainda várias outras características celulares que são essencialmente Booleanas, como o estado de uma proteína quanto a fosforilação ou se uma enzima está em sua forma ativa ou inativa. Isto quer dizer que, apesar de não poderem modelar todo o ambiente celular, as redes Booleanas têm a capacidade de modelar vários aspectos do mesmo fidedignamente.

Existem exemplos conhecidos de regulação genética em que há a presença clara de mecanismos essencialmente Booleanos. Pode-se citar um caso na repressão transcipcional em processos de desenvolvimento celular.

É sabido que a repressão transcripcional está implicada em uma série de processos de desenvolvimento celular, pois apenas o controle positivo não poderia gerar todos os padrões vistos no desenvolvimento. Há duas formas básicas de repressão: a de longo alcance e a de curto alcance, dependendo da distância entre os genes.

No desenvolvimento da *Drosophila melanogaster* (mosca da fruta) a repressão tem um grande papel. Há um gradiente da proteína dorsal de origem materna com um perfil de decaimento muito lento que é responsável pela diferenciação do embrião em três tipos de tecidos embrionários distintos: mesoderma, neuroectoderma e ectoderma dorsal.

Nos embriões desta mosca existem três tipos de promotores com diferentes características que se aproveitam do gradiente da proteína dorsal de diferentes maneiras para causar a diferenciação do tecido embrionário (fig. 3-2).

O promotor de tipo I contém apenas sites de ligação de baixa afinidade à proteína dorsal que só serão ocupados se a concentração desta proteína for alta, como é no caso do mesoderma. Conforme a concentração cai, no neuroectoderma e no ectoderma dorsal, estes sites não são ocupados.

O promotor de tipo II contém sites de ligação de alta afinidade à proteína dorsal. Dado que estes ativadores podem se ligar à proteína dorsal mesmo em baixas concentrações, eles são ligados a ela no mesoderma e no neuroectoderma Entretanto, estes sites também são de alta eficiência para a proteína *snail* que é específica do mesoderma que restringe a expressão destes genes nesta região.

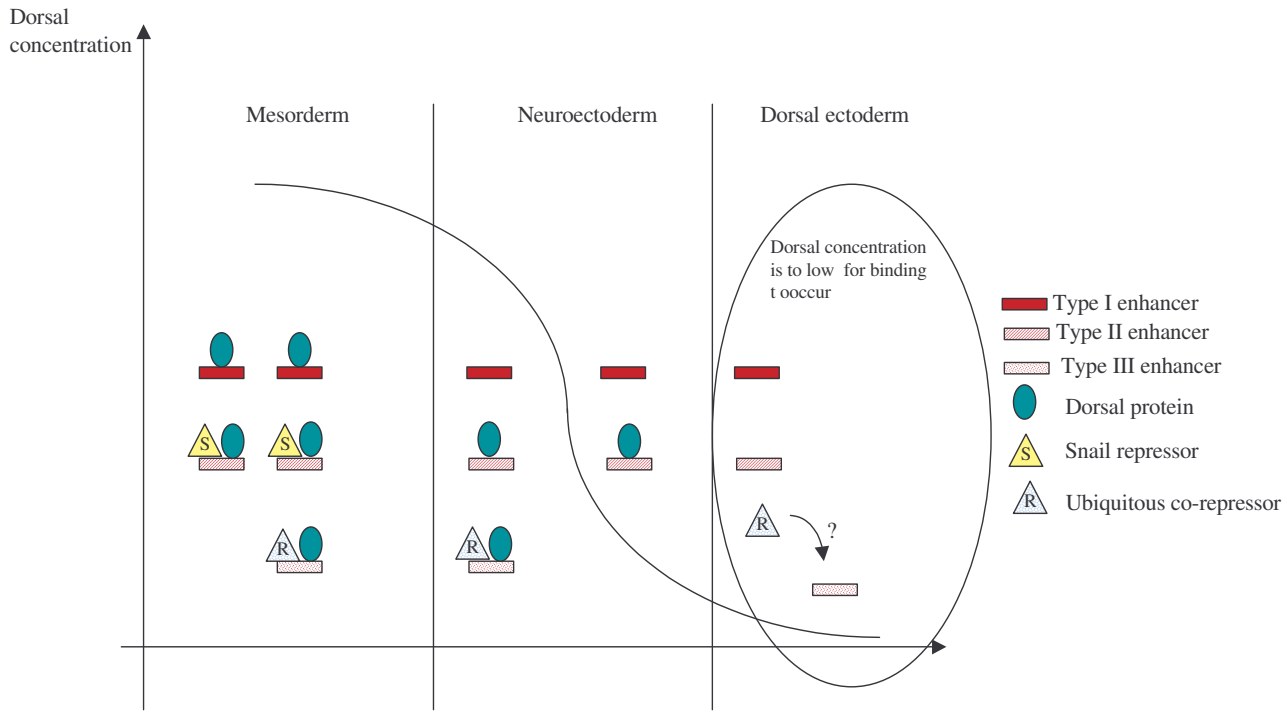


Figura 3-2: Formação de padrões nos embriões pré-celulares da Drosophila melanogaster (extraído de (GRAY et al., 1996)

O último tipo de promotor, de tipo III, contém sites de ligação à proteína dorsal que estão fortemente ligados a sites que se ligam ao co-repressor ubíquo R e nunca são expressos no neuroectoderma e mesoderma. Assim, estes elementos só podem ser expressos no ectoderma dorsal, onde quase não há proteína dorsal para interagir com este co-repressor (GRAY *et al.*, 1996).

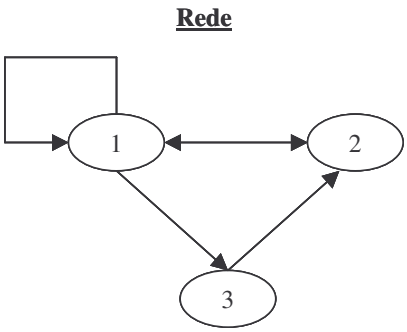
Este exemplo é interessante, pois mostra dois padrões diferentes de controle: um padrão contínuo e um Booleano. No lado contínuo, as diferentes concentrações de proteína dorsal alteram o comportamento dos repressores (caso do neuroectoderma e do mesoderma), enquanto que no padrão Booleano (caso do ectoderma dorsal), a mera presença da proteína dorsal acima de um limite mínimo faz com que a repressão ocorra.

Todos os dois padrões são igualmente importantes. O ectoderma dorsal não se formaria sem o padrão Booleano e os outros dois tecidos não se diferenciariam sem a presença do processo contínuo. Sem qualquer um dos dois, o desenvolvimento da Drosophila seria irremediavelmente alterado.

Este exemplo é um caso típico, que permite que se conclua que a modelagem de certos aspectos da realidade celular através de um mecanismo essencialmente Booleano não

deve ser descartado *a priori* como sendo um modelo não representativo do comportamento de redes de regulação reais.

SZALLASI *et al.* (1998) aplicam o modelo de redes Booleanas ao problema real de modelagem de redes de regulação genética e obtém resultados significativos, mostrando que apesar de redes Booleanas implicarem em perda de informação, eles não implicam em uma perda da compreensão do processo que está sendo modelado naquele momento.



- Função Booleana para o nó 1
  f(1,2)=1 AND NOT 2
- Representação da regulação do nó 1
  1,2,0010
- Significado:

1	2	Next State
0	0	0
0	1	0
1	0	1
1	1	0

Figura 3-3 : Exemplo de codificação adotada nesta tese para a estratégia de regulação para um dos nós de uma rede Booleana. O nó 1 é regulado pelo nó 2 e por um mecanismo de feedback, mas sua codificação só é completa se for conhecido o próximo estado do nó 1 para cada uma das entradas que ele pode receber

3.3 O Modelo proposto

Nesta seção serão descritos os elementos básicos de algoritmo genético aqui proposto, que são a estrutura dos cromossomos, a função de avaliação, os operadores genéticos e os critérios de terminação. O GA definido nesta seção usa o algoritmo básico comum a todos os algoritmos evolucionários, como descrito com detalhes no capítulo 2 desta tese.

3.3.1 Estrutura do Cromossomo

Posto que existem *n* nós na rede Booleana sob estudo, os cromossomos do algoritmo genético aqui proposto serão divididos em *n* partes, denominadas genes. Cada gene armazena a estratégia de regulação completa para o nó *i*, estratégia esta que consiste nos elementos que regulam este nó e a função Booleana que está associada a esta regulação, conforme pode ser visto na figura 3-3.

Assim, cada gene de um cromossomo do GA proposto representa a regulação a que cada nó está submetida no modelo proposto. Assim, pode-se concluir que cada cromossomo no GA representa um modelo de rede de regulação genética proposto como solução para o problema que se está tentando resolver.

Cada bit da função Booleana representa o próximo estado do nó regulado (v_i) dados os estados correspondentes dos nós que o regulam (v_{i1}, v_{i2} , ..., v_{ik}), onde *k* é o número de nós que regulam o nó i, e pode ser diferente para cada um dos nós.

Os bits da função Booleana são representados de forma ordenada, com os nós reguladores servindo como índice para a escolha do bit que informará o próximo estado. O índice é dado pelo número Booleano formado pelos valores correntes dos reguladores, determinando então um número dado por $b_{vi1}b_{vi2}$ ...b_{vik}, onde cada uma das posições é ocupada pelo valor assumido no instante t por um dos nós reguladores do nó v_i.

Assim, o bit armazenado na posição 0 corresponde ao estado no próximo instante do nó v_i se todos os nós reguladores armazenam 0 no instante atual. Da mesma maneira, o valor armazenado na posição 1 corresponde ao estado no próximo instante do nó v_i se todos os nós reguladores armazenam 0 no instante atual, com exceção do nó v_{ik} que armazena o valor 1, e assim por diante.

Seja o seguinte exemplo, do nó v_i regulado pelos nós v_1,v_2 e v_3, e função Booleana 01000010. Os elementos sublinhados na função Booleana são os de índice 1 e índice 6 (considerando que o primeiro elemento é o de índice 0). Logo eles serão selecionados se os nós reguladores tiverem respectivamente os valores b_{v1}=0, b_{v2}=0 e b_{v3}=1 (número Booleano formado é $001_2=1_{10}$) e b_{v1}=1, b_{v2}=1 e b_{v3}=0 (número Booleano formado. $110_2=6_{10}$). A explicação de como calcular o valor do nó *i* no instante t+1 neste exemplo, é dada pela tabela 3-2.

b_{v1} (t)	b_{v2} (t)	b_{v3} (t)	Número formado por $b_{v1}(t)b_{v2}(t)b_{v3}(t)$	v_i(t+1)
0	0	0	$000_2=0_{10}$	0
0	0	1	$001_2=1_{10}$	1
0	1	0	$010_2=2_{10}$	0
0	1	1	$011_2=3_{10}$	0
1	0	0	$100_2=4_{10}$	0
1	0	1	$101_2=5_{10}$	0
1	1	0	$110_2=6_{10}$	1
1	1	1	$111_2=7_{10}$	0

Tabela 3-2 : Valores determinados para o nó i no instante t+1 quando este está sujeito à função de regulação dada por f(1,2,3) → 01000010

3.3.2 Operadores Genéticos

O GA proposto neste capítulo utiliza dois operadores: crossover (recombinação) e mutação, que serão descritos nesta seção em detalhe.

O operador de mutação é aplicado a cada elemento de cada nó, podendo alterar tanto um elemento regulador v_{ik} quanto um bit da função de avaliação através de um mecanismo de sorteio.

Por conseguinte, se for escolhido aplicar o operador de mutação (decisão esta que é tomada de acordo com uma estratégia de seleção de operadores descrita a seguir), um número aleatório *x* é selecionado. Se este número *x* for menor que um valor predefinido a posição em que estivermos naquele momento, seja ela um nó regulador ou um bit da função Booleana, será mudado para um novo valor. Esta operação é consistente com os operadores mais tradicionais de mutação, como vistos em (MITCHELL, 1996) e (FOGEL *et al*., 2003) e pode ser vista em maiores detalhes na figura 3-4.

Nos experimentos conduzidos durante este trabalho, o limite para decisão da alteração da posição corrente de melhor desempenho, entre os vários valores testados, foi

0,5%, que é o valor adotado para todos os experimentos relatados neste capítulo. Isto quer dizer que, uma vez decidida a aplicação do operador de mutação, em média 0,005 dos nós reguladores e dos bits efetivamente sofrerão uma mutação.

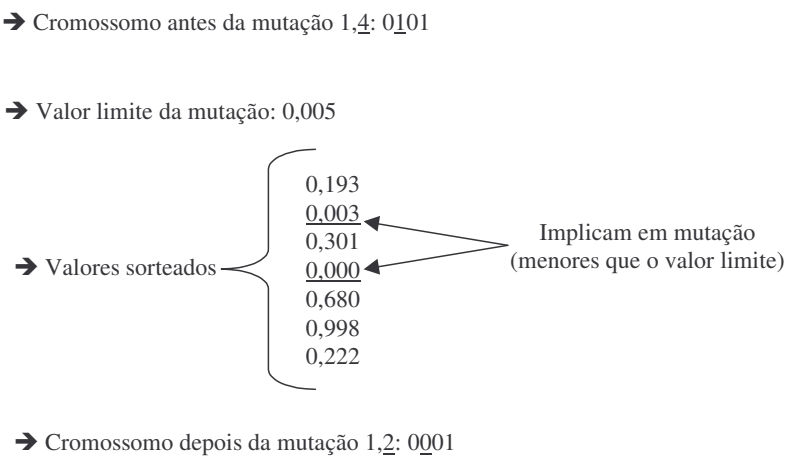


Figura 3-4: Exemplo de atuação do operador de mutação. Sendo o valor limite dado de 0,005, um número é sorteado para cada posição do cromossomo (2 nós e 4 bits da função Booleana). Se algum deles for menor do que o valor do parâmetro, um novo valor é escolhido aleatoriamente para aquela posição (no caso do exemplo, as posições sublinhadas correspondem àquelas que foram mudadas pelo operador).

A estratégia de seleção de operadores consiste na aplicação de uma roleta viciada para os operadores. Isto significa que o algoritmo alterna entre os operadores de mutação e de crossover com uma probabilidade que pode ser fixa ou variável com o tempo (tanto de forma linear como de forma quadrática), dependendo de um parâmetro fornecido pelo usuário.

Nos experimentos mostrados mais adiante verificamos que a estratégia de melhor desempenho é aquela em que o operador de crossover se inicia com uma probabilidade muito alta (95%), probabilidade esta que diminui de forma linear com o passar das gerações, até chegar ao valor de 20% na última geração. A fórmula de cálculo para a probabilidade do operador de crossover é dada por:

$$p_{cross} = 0{,}95 - \frac{0{,}75 * g}{n_{ger}},$$

onde g é a geração atual e n_{ger} é o número de gerações que o GA vai executar. O operador de mutação tem uma probabilidade complementar àquela do operador de crossover, isto é, em todas as gerações temos que:

$$p_{mut} = 1 - p_{cross}$$

A mudança da probabilidade dos operadores deriva do fato de que populações fechadas, com evolução ocorrendo por reprodução entre indivíduos de uma mesma genealogia, tendem a possuir alto grau de convergência genética. Isto quer dizer que no final de uma execução uma população é geneticamente parecida, o que faz com que não seja produtivo cruzar dois elementos, sendo melhor tentar usar mais o operador de mutação para criar maior variabilidade naquela geração.

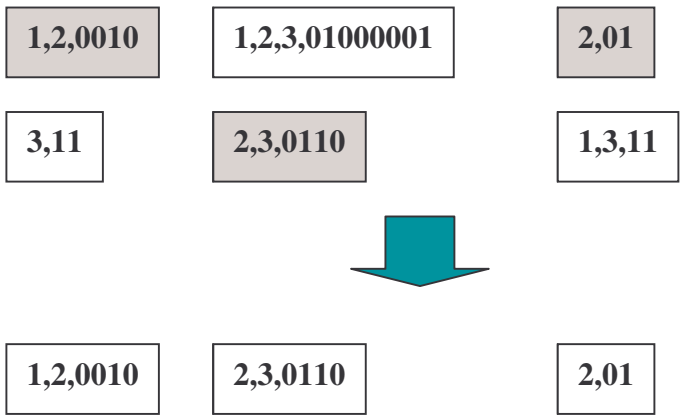


Figura 3-5: Exemplo de operação do crossover aplicado a funções de regulação. Os genes com fundo cinza foram aqueles sorteados para compor o filho gerado pelo operador. O segundo filho seria gerado com os genes marcados com fundo branco. Não há intercâmbio de partes da regulação de cada elemento, mas sim troca da regulação completa.

O operador de crossover é muito simples. Intercambiam-se pedaços de cada um dos indivíduos de acordo com um sorteio. Isto equivale ao operador de crossover uniforme, mas aplicado à regulação completa de cada nó, não aos componentes desta regulação. Isto quer dizer que ao se decidir por trocar as estratégias de um nó, é possível apenas trocar a função completa e não é possível trocar apenas alguns bits da função Booleana, como pode se ver na figura 3-5. Isto significa que o nó v_{ix} no cromossomo x e o nó v_{iy} no cromossomo y não podem intercambiar parte de sua estratégia de regulação (que consiste nos nós reguladores somados à função Booleana). Se o sorteio assim indicar toda a regulação do nó v_{ix} é intercambiada com àquela em v_{iy}, conforme visto na figura 3-5.

3.3.3 Função de Avaliação

Par avaliar a desempenho de um cromossomo, são armazenadas t trajetórias com n_t passos cada. Cada trajetória representa o comportamento desejado (denominada real, de aqui em diante) da rede que se está tentando modelar.

Cada rede recebe o primeiro estado de cada trajetória e são calculados os passos intermediários e final da rede usando a função de avaliação dos nós em questão. O número de bits que diferem em cada um dos passos são somados e o valor final da soma é dividido pelo número de passos.

Esta divisão é necessária se houver mais de uma trajetória para avaliar. Tendo em vista que é possível ter trajetórias mais longas do que outras, se não fosse calculada a média de erros por passo da trajetória, os elementos que acertassem mais bits da trajetória mais longa teriam preferência sobre aqueles que acertam mais bits das trajetórias menos longas. Isto faria com que naturalmente fosse dada preferência àquelas estratégias que acertam em uma das trajetórias (a mais longa), quando na verdade o que se deseja é que todas as trajetórias oferecidas tenham o mesmo peso na função de avaliação.

Esta preferência decorreria do fato de que mesmo poucos bits diferentes por passo em trajetórias longas farão com que se obtenha uma grande soma, enquanto que muitos bits errados por poucos passos levarão a uma soma baixa.

É fácil entender este problema quando vemos um exemplo simples. Imagine que temos duas trajetórias armazenadas para uma rede de 50 bits, a primeira com 11 passos e a segunda com 5 passos. Se tivermos um bit de erro por passo da primeira trajetória, esta somará 11 erros, enquanto que se tivermos dois bits de erro por passo da segunda trajetória, somaremos 10 erros. Logo, se não houver uma divisão pelo número de passos, a contribuição da primeira trajetória para a avaliação deste cromossomo será mais influente do que a contribuição da segunda trajetória, apesar da rede estar bem mais adequada à primeira trajetória. Ao se efetuar a divisão pelo número de passos a situação se inverte e a trajetória com mais erros é a que mais aumenta o valor da função de avaliação.

Posteriormente, cada cromossomo é penalizado de acordo com o número de relacionamentos que ele representa, de forma que os cromossomos mais curtos recebem

créditos extras. Este comportamento se relaciona com o fato de que a natureza normalmente recompensa estratégias minimalistas, pois elas tendem a ser mais curtas, demandar menos energia e serem mais robustas quanto a defeitos na replicação e na transcrição do DNA (ALBERTS *et al*, 2002).

É importante também ressaltar que o número de pontos necessários para identificar corretamente os pesos de uma rede esparsamente conectada é da ordem da conectividade máxima entre os elementos (VAN SOMEREN *et al.*, 2000). Assim, buscando redes menos conectadas é possível extrair mais informação dos dados disponíveis.

Esta penalização é calculada através de um coeficiente de preenchimento. São calculados quantos nós fornecem entradas para o nó i e divide-se este valor pelo número de nós existentes no sistema. Depois calcula-se a média deste coeficiente de preenchimento para todos os nós e será obtido um número que varia de 0 a 1. O objetivo é que a avaliação do cromossomo seja mais penalizada quando este valor está mais próximo de 1, logo diminui-se o valor obtido de 1. A fórmula deste coeficiente é dada, então, por:

$$c = 1 - \frac{\sum_{i=1}^{n} n_i / n}{n},$$

onde n é o número de nós do sistema e n_i, o número de nós que participam da regulação do nó i.

Por exemplo, seja uma rede de três nós cujas regulações são dadas por:

- Nó 1: f(2,3)=0100
- Nó 2: f(3)=00
- Nó 3: f(1,2,3)=11001010

O valor calculado para o coeficiente então é dado por:

$$c = 1 - \frac{2/3 + 1/3 + 3/3}{3} = \frac{1}{3}$$

Este valor é decorrente do fato de que a média de elementos reguladores por nó é igual a 2.

Este coeficiente retorna um valor entre zero e um que será multiplicado pela avaliação do cromossomo obtida da forma descrita anteriormente nesta seção. Assim, se houver dois cromossomos que cometem o mesmo número de erros, aquele que representa uma rede menos conectada passará a ter uma avaliação melhor e será preferido pelo módulo de seleção.

3.3.4 Combinando soluções

Um fator interessante percebido nos resultados preliminares do algoritmo proposto foi o fato de que cada rodada não descobria a regulação correta para todos os nós da rede (aquela que foi usada para gerar os dados fornecidos para o algoritmo), mas sim para alguns destes.

No primeiro dos exemplos citados na próxima seção, apesar de não ser descoberta a regulação correta para a rede completa, as melhores soluções obtidas armazenavam a regulação correta para três dos cinco nós em 72 de 118 rodadas efetuadas (61%), enquanto que duas regulações corretas eram descobertas em 106 das 118 rodadas (90%).

No exemplo número 2 da seção de resultados, as melhores soluções apontadas pelo algoritmo descobriam a regulação correta para cada um dos nós na proporção das rodadas apontadas na tabela 3-3.

Nó	% de acerto
1	62%
2	58%
3	45%
4	35%
5	52%

Tabela 3-3: Percentagem do total de rodadas nas quais foram obtidas as regulações corretas para cada um dos genes do exemplo número 2.

Com base nestes dados decidiu-se acrescentar um passo adicional de combinação dos melhores resultados de nosso GA, de forma a buscar uma solução que contivesse o melhor que cada uma das soluções pudessem oferecer. Foram escolhidas as 8 melhores soluções dadas pelo GA baseando-se apenas no valor da função de avaliação e combinaram-se todas as regulações obtidas.

O limite de 8 soluções foi determinado de forma *ad hoc*. Posto que o espaço de busca cresce de forma exponencial com o número de cromossomos, não se deve determinar um valor muito grande. Ao mesmo tempo, um valor pequeno demais faz com que o propósito do passo de combinação não seja atingido, por não permitir a existência de grande variedade nos cromossomos combinados.

O espaço de busca não é excessivamente grande, apesar da explosão combinatória. Para uma rede pequena, que possua apenas 7 nós, seu tamanho é de 8^7=2,097,152 soluções. Este número de soluções pode ser comparado rapidamente em um computador pessoal (o tempo total de comparação é inferior a 2 minutos em um Pentium IV 1.6Mhz).

Para combinar as soluções simplesmente foram separadas a regulação de cada nó de cada cromossomo que foram posteriormente colocadas em um pool de candidatas a regulação. Obtém-se então um pool para cada nó e as soluções são combinadas. Cada uma das candidatas é avaliada usando-se os mesmos critérios de avaliação descritos na seção anterior e a melhor de todas soluções obtidas é oferecida como a solução encontrada naquela execução do GA.

Este passo de combinação não pode oferecer uma solução pior do que as encontradas pelo algoritmo, dado que as soluções originais do algoritmo são parte do espaço de busca deste passo. Isto implica em que, se as soluções originais forem melhores que todas as outras alternativas, uma das soluções originais terá a avaliação máxima e será retornada pelo passo de combinação. Caso isto não ocorra, a melhor das soluções combinadas será aquela retornada por este passo adicional. Esta possível melhora justifica o esforço computacional dispendido.

3.3.5 Critérios de terminação

O GA usado nesta tese usa três critérios básicos de terminação:

- Qualidade : quando for obtida uma solução que modela perfeitamente os dados, isto é, que prevê todos os bits de todos os passos da trajetória corretamente, então não há necessidade de continuar a evoluir as soluções, pois não há como melhorar os resultados obtidos até aquele ponto.
- Tempo: É estabelecido um limite máximo de gerações para a execução do algoritmo. Se este número for alcançado, mesmo que não tenha sido obtida uma solução ótima, interrompe-se a execução e eventualmente reinicia-se o ciclo de tentativas.
- Estagnação de melhora: Quando se usa elitismo, a avaliação não pode piorar conforme as gerações passam, visto que os melhores indivíduos são levados de uma geração para a outra. Neste caso interrompe-se a execução se não for obtida melhora na avaliação do melhor indivíduo por *n* gerações, onde *n* é um parâmetro fornecido pelo usuário. Este tipo de situação é indicativo de uma forte convergência genética (ausência de diversidade da população) e o que indica que a não ser por uma atuação muito bem sucedida do operador de mutação, não haverá mais melhoras e pode-se encerrar a evolução já que esta tende a não fornecer bons resultados. Uma alternativa interessante consiste em usar o melhor indivíduo como parte da inicialização da população de uma nova execução.

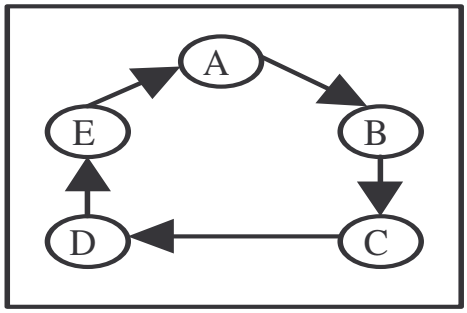


Figura 3-6: Rede usada no exemplo 1, que contém uma configuração cíclica ($A \rightarrow B \rightarrow C \rightarrow D \rightarrow E \rightarrow A$) . A seta quer dizer que o elemento à esquerda da mesma ativa o elemento à direita. Isto é, o elemento à direta assumirá valor 1 no instante t+1 se o elemento à esquerda tiver valor 1 no instante t.

3.4 Resultados

Para testar o algoritmo desenvolvido, foram criados alguns exemplos que apresentam características interessantes ou alguma dificuldade específica. É importante ressaltar que a implementação para este algoritmo foi bastante eficiente e que cada rodada demorou uma média de 5-10 segundos em um computador pessoal padrão (Pentium IV 1.6Mhz), o que significa que é possível executar centenas de rodadas em uma hora.

O tempo requerido para a execução do algoritmo aumenta com o número de nós da rede sendo avaliada e com o número de passos oferecidos para o algoritmo, mas o aumento é linear e afeta apenas a função de avaliação.

O primeiro exemplo utilizado é uma rede de cinco nós que tem uma configuração cíclica, conforme pode ser visto na figura 3-6. A rede contém um ponto fixo (quando todos os nós estão em um estado com valor zero), enquanto que todos os outros pontos iniciais levam a ciclos de tamanho cinco.

É fornecida para a rede apenas uma trajetória contendo 6 pontos. O tamanho da trajetória dada é o tamanho de um ciclo, incluindo o retorno ao ponto inicial. Não é necessário continuar a trajetória, pois tendo em vista que o algoritmo trata apenas redes determinísticas, o valor da configuração do instante seguinte obtido para uma configuração x qualquer será sempre igual, não importando quando esta trajetória ocorra (não há dependência temporal).

A trajetória usada foi *10100* $\rightarrow$ 01010 $\rightarrow$ 00101 $\rightarrow$ 10010 $\rightarrow$ 01001 $\rightarrow$ *10100*. É possível verificar que a última configuração (marcada em itálico) é igual à primeira. Esta informação ajuda o algoritmo na obtenção de trajetórias cíclicas.

O GA foi executado com este exemplo 120 vezes e a avaliação perfeita (isto é, acertamos a previsão de todos os pontos da trajetória) foi obtido duas vezes, o que fez com que não fosse executado o passo de combinação para esta rede, pois não havia como melhorar sua avaliação. As duas respostas perfeitas são mostradas na tabela 3-4.

Casamento perfeito #1
Nó 1 : f(5) → 01
Nó 2 : f(1) → 01
Nó 3: f(2) → 01
Nó 4: f(3) → 01
Nó 5: f(4) → 01
Casamento perfeito #2
Nó 1: f(5) → 01
Nó 2: f(1) → 01
Nó 3: f(2) → 01
Nó 4: f(1,3) → 0111
Nó 5: f(4) → 01

Tabela 3-4 → melhores resultados obtidos para o exemplo #1

Conforme explicado em uma seção anterior, o formato dos resultados significa que, no primeiro caso, a saída do nó 1 no próximo instante é função apenas do nó 5 (f(5)) e seu valor no instante *t+1* é dada por 0, caso o nó 5 cinco seja zero (primeiro bit) no instante *t* e um caso o nó 5 seja 1 (segundo bit) no instante *t*.

Já no caso do segundo resultado, tem-se que a saída do nó 4 é função dos nós 1 e 3 (f(1,3)) e seu valor no próximo instante é dado pela tabela 3-5.

Nó 1 (t)	**Nó 3 (t)**	**Nó 4 (t+1)**
0	0	0
0	1	1
1	0	1
1	1	1

Tabela 3-5 → Interpretação das regras oferecidas como resultado pelo GA

Como se pode ver, na segunda das respostas perfeitas obtidas, o GA forneceu a regulação correta com alguma medida de redundância. Esta redundância consiste em nós

adicionais que não afetam o resultado pois pode-se fazer uma redução simples na função Booleana obtida e chegar à regulação correta.

Outra medida de erro, não visualizada em nenhum dos exemplos citados consiste em obter funções que apesar de obterem resultados perfeitos nos exemplos fornecidos poderiam generalizar mal para outros casos. Isto decorre do fato de que existem infinitas funções Booleanas que obtém os mesmos resultados para o pequeno número de exemplos fornecidos. Infelizmente este é um problema insolúvel, que pode ser minimizado aumentando-se o número de exemplos e aumentando a quantidade de informação fornecida para a rede, conforme discutido a seguir.

Os vários tipos de estimativa de erro dependeriam de quantidades estatisticamente relevante de dados, que não é o caso dos problemas reais que estes exemplos tentam modelar. Quando se trabalha com conjuntos de dados artificiais, pode-se ampliar sem custo a quantidade de exemplos oferecidos para qualquer algoritmo, mas isto não é o caso de problemas biológicos, nos quais a obtenção de cada conjunto de dados custa caro, tanto em termos de tempo quanto financeiro.

Foi feito outro experimento com esta mesma rede, que consistiu em tentar encontrá-la oferecendo três trajetórias de dois pontos cada, descritas na tabela 3-6 a seguir.

11111
11111
00001
10000
01000
01010
00101
10010

Tabela 3-6 : Segundo conjunto de dados oferecido para tentar realizar a inferência da rede de regulação do exemplo #1

Neste exemplo, tem-se então o mesmo número de medidas, mas com um número diferente de estados iniciais. Em 40 execuções do algoritmo, foi obtido o escore perfeito 12 vezes, 8 das quais incluíam redundâncias similares àquelas descritas acima.

Estes resultados sugerem que um número maior de medidas com uma única condição inicial contêm menos informação do que muitas trajetórias com várias condições iniciais distintas. Este fato já foi sugerido por D'HAESELEER (1993), e implica em que a abordagem de se investir um grande esforço na obtenção de uma única trajetória longa não é muito produtiva, devendo-se privilegiar a possibilidade de se estudar mais trajetórias, mesmo que mais curtas, mas com estados iniciais diferentes.

A questão consiste em que para inferir a regulação de um único gene, tende a ser mais produtivo observar a expressão do mesmo em várias diferentes combinações de níveis de expressão de suas entradas. Neste caso, têm-se amostras de várias condições e perturbações das condições ambientais.

As séries temporais de expressão gênica oferecem uma grande quantidade de dados, mas todos os pontos tendem a ser sobre um único processo dinâmico na célula e estarão fortemente correlacionados aos pontos nos instantes anterior e posterior. Logo, espera-se que uma série temporal de *n* pontos tenha menos informação do que *n* pontos medidos de forma independente, em condições ambientais bem distintas. As séries podem, entretanto, prover uma boa visão da dinâmica do processo como um todo, além de serem uma conseqüência natural de se obter dados através da tecnologia de microarrays que será discutida mais adiante nesta tese.

Obviamente, os dois tipos de dados não são mutuamente excludentes, e quanto mais se obtiver de cada um deles, mais preciso será o modelo obtido e mais próximo da realidade o resultado do algoritmo aqui proposto.

Foi oferecido ao algoritmo um segundo exemplo que consistia em uma rede dada pela configuração descrita na tabela 3-7. Este exemplo é mais complexo, tanto por possuir mais nós quanto por possuir um grau maior de interconexão entre os elementos.

O_1 : f(2,3) → 0111
O_2 : f(1,3,4) → 01101000
O_3 : f(3) → 01
O_4 : f(4) → 10
O_5 : f(3,4) → 0111
O_6 : 0
O_7 : f(1,2,3,4,5) : 00001111000000001111001010101010

Tabela 3-7: Rede de regulação real a ser descoberta no exemplo #2.

Esta rede contém sete nós, dos quais um deles é constante (o nó 6), mantendo-se permanentemente não expresso (com valor zero). Há dois nós que recebem uma entrada (on nós 3 e 4), dois que recebem duas entradas (1 e 5), um que recebe três entradas (nó 2) e um que recebe 5 entradas (o nó 7). A compreensão desta rede é facilitada pela representação em formato de grafo, como pode ser visto na figura 3-7.

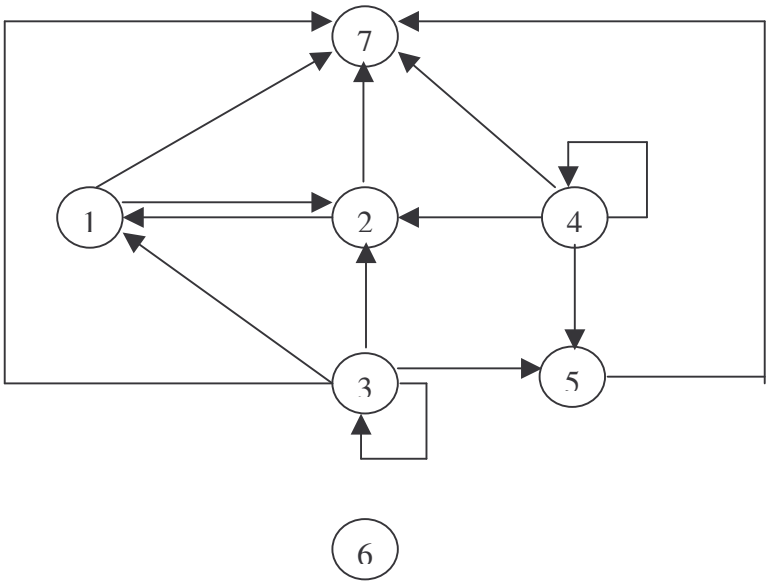


Figura 3-7: Representação do exemplo com regulação dada pela tabela 3-7 em formato de grafo. As setas indicam que o nó de origem regula o nó de destino.

Este último nó demonstra o principal problema da nossa representação que consiste no tamanho da *string* necessária para representar uma função. O número de bits necessários obviamente é dado por 2^n, onde n é o número de nós reguladores. Este número cresce

exponencialmente e pode ser problemático em casos de redes grandes e fortemente interconectadas.

Este problema foi corrigido no programa genético usado no caso de redes contínuas através do uso de uma representação distinta. O modelo proposto no capítulo a seguir é que é muito mais compacto, além de poder lidar também com exemplos Booleanos.

Foi executado o GA oferecendo-lhe a seguinte trajetória:

1101000 → *1000101* → 0101001 → 1100100 → 1101000 → *1000101* → 0101001 → 1100100 → 1101000 → *1000101*

Esta trajetória consiste em 10 pontos, sendo que o último ponto (em itálico) consiste em uma repetição dos pontos número 2 e 6 (também em itálico), indicando a existência de um ciclo atrator.

Foram executadas 20 rodadas do GA proposto e em nenhuma delas foi encontrada a rede correta. Em todas as rodadas foi descoberta uma rede que terminava exatamente no mesmo atrator da trajetória exemplo

Em 12 das 20 rodadas (60%) a rede descoberta pelo GA calculava um resultado que diferia da trajetória exemplo em 3 bits ou menos (de um total de 63, dado que o primeiro passo não é calculado, mas sim dado para a rede como valor de inicialização), correspondendo a um erro de 5% ou menos. Isto é esperado, dada a ambigüidade inerente de um conjunto de dados tão pequeno. A melhor rede encontrada, após o passo de combinação, pode ser vista na tabela 3-8

Nó 1 : f(2,3,5) → 10011101
Nó 2: f(1,3,4,6) → 1011001110011011
Nó 3: 0
Nó 4: f(4) → 10
Nó 5: f(3,4) → 0110
Nó 6: 0
Nó 7: f(1,2,3,4,5) : 11110001100100010100001000111000

Tabela 3-8 : Melhor rede de regulaçao encontrada para o exemplo #2.

Esta rede não contém a regulação correta, mas ela oferece alguns resultados interessantes que devem ser analisados.

Primeiramente, a melhor regulação para o nó 3 consiste em concebê-lo como um nó constante em zero. Isto decorre do fato do exemplo fornecido para o algoritmo só conter pontos nos quais este nó assume valor zero. Logo, o fato de haver apenas uma trajetória prejudica o algoritmo proposto (e qualquer outro que porventura buscasse extrair informações de tal conjunto de dados).

Segundo, o algoritmo proposto obteve, para os nós 1 e 2, os corretos elementos reguladores com a adição de mais um nó espúrio e os elementos regulatórios corretos para os nós 4,5,6 e 7, apesar de não acertar todos os bits da função Booleana.

Entretanto, estes bits não forem importantes na avaliação da trajetória usada como exemplo, pois a rede descrita acima termina no ciclo atrator especificado no exemplo no caso do exemplo dado. Se houvesse outras trajetórias, a pressão evolucionária poderia fazer com que outros bits também convergissem para os valores corretos.

Além disto, dado que a função Booleana é previamente conhecida, decidiu-se verificar como a solução descoberta se comportaria para os $2^7 = 128$ possíveis trajetórias de tal rede. Pode-se verificar então que ela calculou 26 (20,3%) das trajetórias corretamente e das outras 102, 85 (83,3%) diferiam das trajetórias "reais" em menos de um bit por passo.

Outros exemplos foram fornecidos para o algoritmo proposto. Um deles foi uma simulação do operon Lac nas bactérias E. Coli que é um mecanismo de controle para as enzimas necessárias para a digestão das lactoses, sendo um exemplo interessante tanto para mecanismos de repressão e ativação genéticos.

Como discutido na seção 2.3, o dogma central da biologia estabelece que o DNA primeiro precisa ser transcrito em uma substância intermediária, o RNA mensageiro (mRNA) antes de ser traduzido para uma proteína. Isto implica em que os níveis de proteínas podem ser regulados nos passos de transcrição de DNA em mRNA ou no passo de tradução deste em proteína. Esta regulação está relacionada com a necessidade da célula de reconhecer e responder a novas condições ambientais.

O controle da transcrição é dado através da ligação de repressores ou indutores a um sítio do DNA denominado operador. Este sítio se localiza próximo ao gene codificador da

proteína. Este conjunto operador mais proteína codificadora é denominado operon, e constitui a unidade básica de coordenação da expressão gênica (GRIFFITHS, 2000).

O operon lac é composto de três genes estruturais: um que codifica para a β-galactosidae (chamado z), um que codifica para a permease (chamado y) e outro que codifica para a transacetilase (denominado a). Este último não está envolvido na digestão da lactose, mas é sempre co-expresso com os outros dois.

O operon inclui um operador que se liga a uma proteína codificada pelo gene denominado l que está codificado logo acima na cadeia de regulação (Fig 3-8). Quando está proteína está ligada, o operon é reprimido e a DNA polimerase, mesmo quando ligada à região promotora (P), não consegue prosseguir com o processo de transcrição. Assim, podemos dizer que o operon lac está submetido a um controle negativo por esta proteína, que pode então ser chamada de repressora.

Entretanto, quando a lactose está presente, ela se liga a esta proteína repressora, desativando-a, o que permite que o processo de transcrição das enzimas necessárias para a sua digestão prossiga.

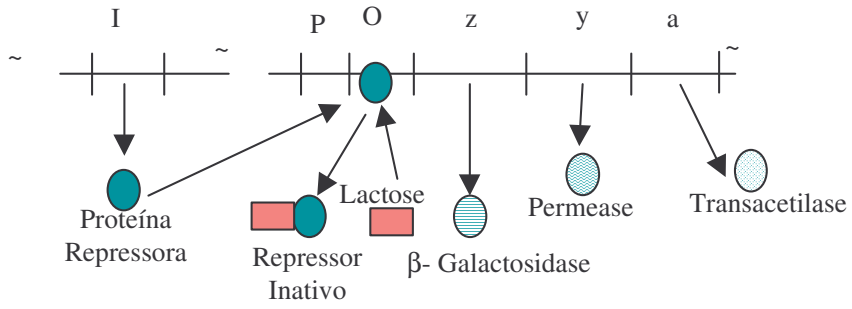


Figura 3-8: Estrutura do gene codificador do operon Lac.

O operon lac também está sob controle positivo. A proteína CAP (Catabolite Activator Protein, ou Proteína Ativadora de Catabólitos), que é codificada pelo gene crp situado bem mais longe do que a região codificadora da proteína repressora, se liga ao cAMP para formar um complexo que se liga ao promotor lac e facilita sua transcrição, fazendo com que mais cópias das proteínas codificadas pelos genes lac estejam disponíveis.

Entretanto, as bactérias E. Coli preferem a glicose como fonte de energia, em detrimento da lactose. Isto fica claro quando analisamos o fato de que quando a glicose está

presente, a quantidade de cAMP é diminuída através de várias reações químicas, fazendo com que haja menos complexos CAP-cAMP e ocorra uma diminuição do número de genes expressos.

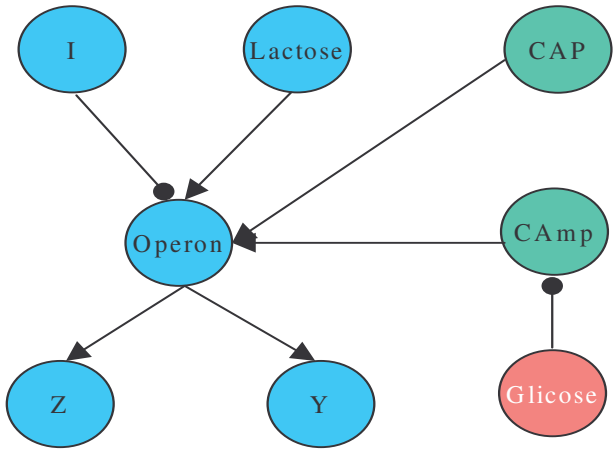


Figura 3-9: Rede usada para a simulação do open Lac. As setas simbolizam uma atuação positiva (ativação) enquanto que os cícrculos indicam repressão.

Foi então criada uma rede Booleana (mostrada na fig. 3-9) buscando representar estes relacionamentos de forma fiel. Foram gerados então dados sintéticos para que o algoritmo proposto pudesse fazer a inferência desta rede.

Foram criadas três trajetórias que terminam em atratores simples. Todos os dados possíveis foram gerados (2^6 trajetórias) e verificou-se que a rede não apresenta nenhum tipo de ciclo ou qualquer outro atrator relevante. A seguir pode-se ver as trajetórias oferecidas para a rede foram as seguintes, informando que os bits estão ordenados da seguinte forma: Proteína Repressora (I, na figura), que consideramos presente quando está em sua forma ativa, Lactose, Glicose, CAP, CAMP e Operon.

Para esta rede, foi obtido o resultado mostrado na tabela 3-10, após a aplicação do algoritmo proposto, após apenas 10 rodadas.

A ordenação dos bits é dada da maneira descrita anteriormente, com o nome que aparece na j-ésima posição substituindo o nó v_{ij}. Por exemplo, para a regulação da Lactose tem-se a ordenação dos nós reguladores, para efeito de cálculo do valor da Lactose no instante t+1, dada pela tabela 3-11.

100110
100110
110110
110111
110111
111110
111101
111100
111100

Tabela 3-9: Trajetórias oferecidas para o algoritmo para a inferência do mecanismo do operon lac.

I : f(Lactose) → 10
Lactose : f(Lactose,CAP) →0010
Glicose : f(RP,Lactose,Glicose) → 00110001
CAP : f(Glicose,Operon) → 1000
cAMP : f(RP,Glicose,Operon) → 11001100
operon : f(RP,Lactose,Glucose,cAMP,Operon) → 0101100001000100100011110011010 0

Tabela 3-10: Melhor resultado obtido pelo algoritmo na inferência da rede de regulação do operon lac

Lactose(t)	CAP(t)	Lactose(t+1)
0	0	0
0	1	0
1	0	1
1	1	0

Tabela 3-11: Forma de calcular o valor da Lactose no instante t+1, dados os valores de seus nós reguladores. Os reguladores são ordenados conforme aparecem na definição da função de regulação explicitada na tabela 3-10.

Estes resultados apresentam aspectos interessantes:

- A proteína repressora (resultado da expressão do gene I), é controlada da Lactose está presente apenas se esta está ausente, o que é exatamente igual ao comportamento real desta substância.
- A lactose é função dela mesma e do CAP. Este tipo de resultado trivial (a presença de um elemento estar relacionado com a presença dele mesmo no instante anterior) é uma possibilidade dentro de qualquer algoritmo, visto que um elemento costuma ter alta correlação com ele mesmo. Entretanto, neste caso pode-se verificar que quando CAP está presente (bits 1 e 3) a lactose some no instante *t+1*. Isto é aproximadamente o que acontece no operon, tendo em vista que o CAP realiza um controle positivo sobre as duas enzimas necessárias para a digestão da lactose. Logo, sua presença poderia indicar que a lactose seria digerida em um instante de tempo próximo.
- O CAP, de acordo com o algoritmo proposto, está ausente quando a glicose está presente (bits 2 e 3 da função de controle), o que também corresponde, a grosso modo, ao controle real do operon Lac. O mesmo vale para o caso do cAMP, cujos bits 2,3 6 e 7 correspondem ao caso em que a glicose está presente (respectivamente 0$\underline{1}$0, 0$\underline{1}$1, 1$\underline{1}$0 e 1$\underline{1}$1, onde a presença de glicose é dada pelo segundo bit) e indicam que no instante *t+1* não haverá a presença de cAMP.

Pode-se concluir então que vários aspectos reais do controle realizado no operon Lac foram modelados de forma bem sucedida por este algoritmo, mesmo com um número reduzido de rodadas.

Estes resultados são encorajadores, mesmo considerando que as rodadas do GA não retornaram as redes de regulação corretas. É importante compreender que a quantidade de informação fornecida para o GA é muito pequena e que toda trajetória que não pertence ao nosso conjunto de treinamento deve ser extrapolada, o que significa que a restrição colocada sobre os dados é pequena. Várias funções Booleanas têm o mesmo resultado em um único ponto, o que significa que qualquer uma delas seria considerada como um acerto pela função de avaliação utilizada.

Por exemplo, seja o caso de se prever o valor da Lactose no instante t+1, posto que no instante t temos os seguintes valores para cada um dos elementos envolvidos no processo:

- Proteína Repressora:1

- Lactose:0
- Glicose:0
- CAP:1
- CAMP:1

Se o valor da lactose no instante t+1 for 0, dentre as várias funções que acertam o valor, pode-se apontar as seguintes:

- NOT(Proteína Repressora(t-1))
- Proteína Repressora(t-1) AND Glicose(t-1)
- Glicose(t-1)
- Lactose(t-1) AND CAP(t-1)
- 0

Existem muitas outras funções capazes de acertar perfeitamente o padrão, e, na ausência de qualquer outra informação adicional, qualquer uma delas é aceitável como resposta do algoritmo.

Quando existe mais de um ponto ou mais de uma trajetória, este problema diminui, mas não desaparece por inteiro. Sempre haverá mais de uma função lógica capaz de combinar os elementos presentes no estado no instante *t-1* de forma a prever os elementos do estado no instante *t*.

Quando se está modelando redes possivelmente complexas em uma situação em que há poucas trajetórias, enfrenta-se uma situação em que apenas um pequeno subconjunto de todas as possibilidades de relacionamento entrada/saída de cada função Booleana está presente. Assim, qualquer função que acerte estes poucos bits sem erro obterá um escore altíssimo no algoritmo, mesmo que não extrapole os outros bits de forma correta.

Esta falta de restrição claramente visível no exemplo número dois visto acima. O bit 3, por acaso, permaneceu constante em zero em todos os pontos da trajetória. Isto fez com que o GA calculasse como casamento perfeito uma função constante (k=0) como regulação para este nó.

É sabido, posto que o exemplo foi criado artificialmente, que tal regulação não é a verdadeira, mas, tendo em vista que nenhum tipo de informação adicional foi fornecida, o

GA não tem como saber isto, da mesma maneira que qualquer outro método computacional que porventura viesse a ser aplicado.

Este tipo de ambigüidade é inerente às condições em que os dados genéticos reais são obtidos, como será visto no capítulo 4 desta tese. Dado o alto custo de se medir o padrão de expressão de vários genes, o número de medidas é significativamente menor que o número de genes (às vezes, até 3 ordens de grandeza menor). Logo, haverá um grande número de soluções que se encaixam nos dados disponíveis (VAN SOMEREN *et al.*, 2000).

A solução para este problema é, obviamente, incluir mais trajetórias exemplo. AKUTSU *et al.* (1999) sugere um número e uma estrutura de exemplos que representam uma boa maneira de aumentar o volume de informação dada para algoritmo evolucionário proposto nesta tese. Neste artigo é demonstrado que o número de entradas necessárias para uma rede ser determinada é de O(log n), quando o número de relacionamentos (K) entre elementos é fixo.

O problema de tal abordagem é que, como o próprio artigo aponta, em situações reais é impossível controlar os valores oferecidos pelo algoritmo, pois os dados são obtidos como passos consecutivos de uma trajetória, o que pode fazer com que sejam enfrentados problemas como aquele vistos no caso do exemplo dois mostrado acima. Neste, um dos elementos permaneceu constante, não oferecendo qualquer informação para que o algoritmo realizasse uma inferência.

O problema é de fácil solução quando se lida com dados sintéticos – basta gerar mais dados para o algoritmo. Entretanto, quando se discutem experimentos em que a geração de dados implica em um custo real, nem sempre tal ação será possível. Logo, os algoritmos devem procurar realizar o melhor trabalho possível com um volume escasso de dados.

Outro problema sério da abordagem apresentada em (AKUTSU *et al.*, 1999) é que as constantes associadas ao método crescem de forma exponencial com o valor máximo de interconexão entre os elementos, representado pelo símbolo K. Ademais, no trabalho descrito naquele artigo usam-se apenas valores de K=2 e K=3, quando já foi demonstrado que na natureza os valores reais de K oscilam entre 4 e 8 (ARNONE *et al.*, 1997).

Não se procurou aprofundar este ponto pois este algoritmo mostrou-se frágil, tendo em vista o fato de seus cromossomos crescerem de forma exponencial com o tamanho e a interconexão dos nós da rede. Logo, seria interessante busca uma representação alternativa, que será mostrada no próximo capítulo.

Para exemplificar este problema, foi criado um exemplo com 30 nós, para os quais foram fornecidas duas trajetórias distintas de 15 passos cada. O número médio de interconexões entre os elementos desta rede consistia em cerca de 3. Com isto, o cromossomo que representava a configuração correta tinha 3570 caracteres. Em situações reais nós podemos ter até 8000 genes em um exemplo (como visto no caso do capítulo 4 desta mesma tese), o que levaria o cromossomo a ter um tamanho que não seria representável com uso adequado de memória.

Entretanto, os resultados, até mesmo para uma rede relativamente grande como esta de 30 nós são bastante animadores. Após rodar o GA 50 vezes, com população de 100 indivíduos e estratégia elitista, os dois melhores indivíduos de cada rodada foram usados para inicializar a população da 51ª rodada que retornou um resultado que acertava 585 de 840 bits em teste (aproximadamente 70%). É importante ressaltar que o primeiro passo de cada trajetória é dado como ponto inicial para a função de avaliação, o que faz com que não seja incluído no cálculo dos acertos.

O mais interessante é que das 30 regulações, 3 foram descobertas corretamente e 2 contendo redundâncias como as descritas anteriormente (16% de acerto real). Das 25 restantes, 8 continham (32%) exatamente os elementos que eram os reguladores corretos, mas os bits descobertos estavam incorretos.

Estes resultados são alentadores, mas insuficientes para justificar esta representação. Ainda é necessário algo mais compacto, para que seja possível tratar problemas mais próximos da realidade. Entretanto, se for possível compartimentalizar as redes de regulação através de um algoritmo de particionamento de grafos, como descrito por KAWAJI *et al.* (2001), então pode-se encontrar utilidade para este algoritmo.

3.5 Conclusão

Neste capítulo foi mostrada a aplicação de um algoritmo evolucionário para a inferência de uma rede Booleana que pudesse, entre outras aplicações, modelar certos aspectos do desenvolvimento e da vida de uma célula e da regulação da expressão gênica.

O algoritmo genético utilizado demonstrou possuir uma boa capacidade para encontrar redes Booleanas ajustadas aos dados oferecidos para o treinamento, encontrando inclusive as regulações corretas em vários dos casos que lhe foram apresentados bem como modelando certos aspectos críticos do comportamento das redes apresentadas, como pontos e ciclos atratores.

Entretanto, a representação cromossomial utilizada apresenta problemas de ocupação eficiente de memória e de uso quando a rede apresenta um número maior de nós. Este problema não é derivado de uma implementação específica, mas sim inerente ao fato de que para representar a função de regulação de um nó que interage com n outros nós, são necessários 2^n bits. Isto inviabilizaria a aplicação deste algoritmo em problemas biológicos reais, nos quais o número de nós pode chegar às dezenas de milhares.

Ademais, apesar das redes Booleanas demonstrarem ser capazes de modelar certos aspectos interessantes do desenvolvimento celular, elas apresentam limitações que devem ser tratadas quando se deseja obter modelos mais precisos. Estas limitações são mais evidentes quando é levantada a questão das respostas graduais, existentes no ambiente celular tipicamente sob o formato de funções sigmoidais. Estas respostas são aproximadas por funções degrau em redes Booleanas, funções estas que não apresentam a mesma gradualidade de resposta.

Assim, torna-se necessário buscar um modelo que possua uma representação mais compacta, de forma que se possa lidar com problemas maiores, e que seja capaz de lidar com variáveis contínuas, de forma a preservar a gradualidade de resposta, o que será feito em detalhes no próximo capítulo.

Capítulo 4 - Algoritmo Contínuo Proposto

4.1 Introdução

Neste capítulo será descrito em detalhe o algoritmo que foi desenvolvido para modelar processos contínuos. Serão mostrados cada um dos passos que são necessários para que sejam obtidos os resultados demonstrados nos capítulos posteriores.

Como visto no capítulo anterior, a modelagem através de redes Booleanas é interessante como modelo simplificado da realidade, possuindo a capacidade de refletir diversos aspectos da realidade de forma relativamente precisa. Entretanto, existem limitações nesta abordagem que nunca serão transpostos, como a incapacidade de modelar o controle gradual.

Dadas estas limitações, e os problemas encontrados no algoritmo proposto (como o tamanho excessivo dos cromossomos), decidiu-se partir para um algoritmo mais sofisticado, que pudesse modelar tanto dados contínuos quanto Booleanos, algoritmo este que será desenvolvido neste capítulo.

Este algoritmo consiste na utilização de um algoritmo evolucionário (EA) para a descoberta de regras fuzzy que modelem o sistema de controle que gerou os dados sob análise.

A idéia de representar redes de regulação como um conjunto de regras não é nova. Ela já foi explorada em vários trabalhos anteriores. MCADAMS *et al.* (1995) comparou a dinâmica celular com a dinâmica de circuitos e simulada através do uso de regras representáveis como circuitos lógicos. VENET *et al.* (2001) define chaves binárias usando-as como o mecanismo fundamental no qual se baseiam regras binárias que são usadas para descobrir relacionamentos úteis na regulação genética.

Para desenvolver as regras procura-se concentrar em uma ou mais variáveis de interesse presentes no conjunto de dados oferecido para o algoritmo. Estas variáveis também são chamadas de elementos de interesse, ou elementos, ao longo desta tese.

Estes elementos de interesse são determinados de acordo com a necessidade do usuário. Por exemplo, em uma aplicação de classificação a variável de interesse poderia ser a categoria a que a tupla pertence, enquanto que em uma aplicação de engenharia reversa de redes de regulação genética a variável de interesse pode ser o nível de expressão de um gene específico.

Para cada uma delas tentará se descobrir, através do uso do EA, um conjunto de regras fuzzy de forma que os valores calculados através da aplicação destas regras seja uma aproximação o mais próximo possível ao valor presente nos dados (chamado de valor real).

Quando se está falando de um exemplo de bioinformática, diz-se que as regras fuzzy representam um modelo da regulação genética e que o conjunto de regras fuzzy é um modelo da rede de regulação genética. Assim, o conjunto de regras fuzzy é denominado de rede de regulação (ou simplesmente de rede) durante este capítulo.

Como discutido no capítulo dois desta tese, um algoritmo evolucionário funciona iterando uma população de estruturas, denominadas indivíduos, que são evoluídos em busca da solução final do problema. No caso do EA usado neste capítulo, cada indivíduo consiste em uma rede de regulação completa, contendo uma ou mais regras para cada uma das variáveis de interesse.

Regras fuzzy se baseiam em conjuntos fuzzy definidos de acordo com a compreensão do fenômeno sendo modelado. Conforme visto no capítulo dois, os conjuntos

fuzzy representam a divisão dos valores que as variáveis podem assumir (seu universo de discurso) através do uso de funções que modelam a pertinência da variável a um conceito lingüístico determinado. A definição dos conjuntos fuzzy usados para criar as regras evoluídas pelo EA será discutida na seção 4.2 a seguir.

Cada uma das regras que compõem uma rede de regulação consiste em uma expressão composta de um ou mais conjuntos fuzzy aplicados às variáveis do conjunto unidos por conectivos lógicos, de forma a gerar expressões de qualquer grau de complexidade, conforme se discutirá na seção 4.2 deste mesmo capítulo.

Cada indivíduo, que consiste em uma rede (ou base de regras) completa, será avaliado de acordo com uma função de avaliação discutida em detalhes na seção 4.3 deste capítulo. Esta função mede a qualidade desta rede como solução do problema, podendo ser definida de diversas maneiras de acordo com a aplicação que se tem em mente. No caso de uma aplicação de análise de microarrays a função de avaliação calcula quão bem a trajetória real do gene de interesse foi modelada.

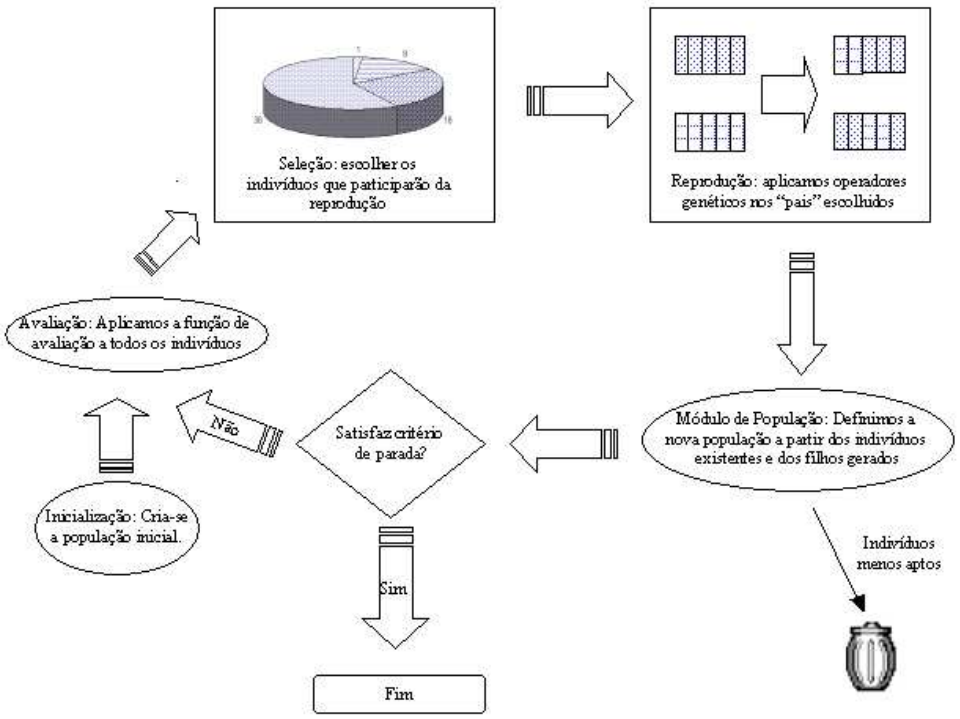


Figura 4-1 → Esquema do EA desenvolvido nesta tese.

Esta função de avaliação será usada como base para a seleção dos indivíduos para aplicação dos operadores genéticos. Estes operadores determinam como um EA combina e altera soluções de forma a evoluí-las, gerando outras redes, denominadas descendentes. Os operadores usados neste algoritmo são descritos na seção 4.4 deste capítulo.

Finalmente, os descendentes e os indivíduos que os geraram (população no instante *t*) são submetidos a um módulo de população, que determina todos aqueles que devem sobreviver, isto é, que devem compor a população no instante *t+1*.

Este algoritmo é baseado no esquema geral de algoritmos evolucionários discutido no capítulo 2 desta tese, esquema este que foi concebido conforme diagrama mostrado na figura 4-1. Como é possível ver na figura, é necessário definir como são selecionados os indivíduos a participar da população, os operadores genéticos, o módulo de população usado e o critério de parada.

Pode-se ver no diagrama que também é necessário que se defina uma função de avaliação. Esta função é específica a cada problema para o qual se aplicará este EA e para determiná-la é necessário conhecimento específico da área na qual se aplica o algoritmo evolucionário.

No caso da aplicação descrita neste capítulo descreve-se a função de avaliação usada para estudos de previsão e engenharia reversa de redes de regulação gênicas. Outra aplicação possível (classificação) é descrita no apêndice A desta tese e as mudanças necessárias na função de avaliação são descritas de forma detalhada naquele capítulo.

Para acelerar a obtenção de resultados foi aplicada uma técnica de paralelização de algoritmos evolucionários denominada *island*. Esta técnica permite que se executem múltiplas instâncias de um EA de forma assíncrona, com uma comunicação entre elas. Isto permite que se desenvolva uma população maior e ao mesmo tempo mais diversificada, o que acarreta na obtenção de resultados melhores mais rapidamente, diminuindo o número de rodadas do algoritmo necessárias. Esta técnica de paralelização é descrita de forma detalhada na seção 4.5 deste capítulo.

As próximas seções discutirão de forma detalhada cada um dos elementos básicos mencionados acima, de forma que o funcionamento do algoritmo evolucionário desta tese fique claro.

4.2 Estrutura do cromossomo

4.2.1 Conceitos

Conforme discutido na introdução, cada cromossomo (ou indivíduo) da população mantida pelo algoritmo proposto nesta tese representa uma base de regras fuzzy. Estas regras podem ser aplicadas ao problema sendo estudado de forma a verificar se foi obtido o resultado desejado. Esta aplicação é feita pela função de avaliação, descrita detalhadamente mais à frente, na seção e) deste capítulo. Nesta seção será descrita a representação das regras pertencentes à base fuzzy representada pelo cromossomo.

Uma base de regras consiste em um conjunto de regras no seguinte formato:

SE <antecedente> ENTÃO <conseqüente>

O <conseqüente> de cada uma destas regras consiste em um conjunto fuzzy, enquanto que os antecedentes de cada regra podem ser descritos como uma expressão em notação polonesa reversa (RPN).

Nesta notação, todos os operadores antecedem seus operandos, que por sua vez podem ser uma expressão que possui seu próprio operador. A definição sintática de uma expressão em RPN é dada pelas seguintes regras, descritas usando-se notação BNF:

<expressão> ::= AND <expressão> <expressão>| OR <expressão> <expressão>|NOT <expressão> | <operando>

onde <operando> é o elemento fundamental da expressão. No caso do algoritmo desta tese, este elemento fundamental é um conjunto fuzzy, como se verá mais adiante nesta seção.

Como pode ser visto ver nesta definição, são usados apenas três operadores: NOT, OR, AND. Apesar de ser possível representar toda e qualquer operação lógica usando apenas dois deles (NOT e AND ou NOT e OR), o uso de três operadores simplifica as expressões presentes no antecedente de cada regra permitindo que as regras sejam mais curtas.

O fato de serem usados apenas três operadores não gera nenhum tipo de limitação do poder de representação de regras dos cromossomos utilizados. A combinação de operações geradas com os operadores usados aqui permite a representação de qualquer expressão lógica desejada.

A RPN é adequada para uma representação em formato de árvore em que os descendentes são as sub-expressões que formam uma árvore enraizada no operador. Cada operador tem dois descendentes, com exceção do operador NOT, que possui apenas um (o que deriva do fato deste ser o único operador unário dentre aqueles usados). Esta representação em árvore é adequada para uma apresentação gráfica, como pode ser visto na figura 4-2.

YANG *et al.* (2003) e DASGUPTA *et al.* (2002) já usaram a notação polonesa, porém com diferenças significativas em relação à utilização neste trabalho. As devidas comparações são feitas na seção 4.7 desta tese.

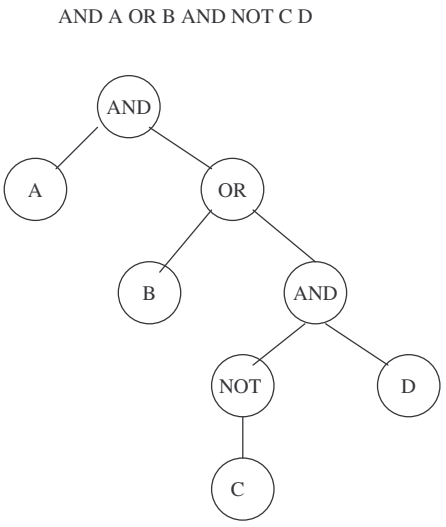


Figura 4-2 → Representação em formato de árvore de um conseqüente de uma regra em formato de RPN

A compreensibilidade das regras para um usuário humano é uma característica desejável do conhecimento descoberto pelo algoritmo aqui descrito. Isto é necessário sempre que se utiliza o conhecimento obtido para apoiar uma decisão tomada por um ser humano (FREITAS, 2000).

Dado que o propósito principal das aplicações descritas nesta tese consiste em criar hipóteses que serão fornecidas para investigadores checarem em bancada (capítulo 5) ou para utilização em aplicações de classificação (capítulo 6), a simplicidade das regras é um dos objetivos que o algoritmo procura maximizar, o que implicou em mudanças tanto no algoritmo de inicialização de regras quanto na função de avaliação (que será descrita na seção 4.3 neste mesmo capítulo).

A questão da simplicidade é relativa, dependendo da capacidade e do ponto de vista do usuário da regra, o que faz com que seja especialmente difícil classificar uma regra quanto a sua simplicidade. Assim, é necessário adotar um critério objetivo que seja fortemente ligado a este conceito abstrato. Decidiu-se nesta tese por buscar uma menor altura das árvores de expressão, isto é, por regras mais curtas.

No caso do gerador de funções, isto foi atingido através da criação de um parâmetro no mecanismo de inicialização das regras denominado *coeficiente de altura de expressão*, cujo objetivo é fazer com que se gerem árvores mais baixas. A idéia é que este coeficiente seja inversamente proporcional à altura média das árvores criadas, de modo que regras mais complexas, que por conseguinte têm representação em formato de árvore mais altas, sejam punidas, dando-se preferência a árvores de altura menor. Este coeficiente de altura será explicado com mais detalhes na seção sobre implementação (4.2.2) deste mesmo capítulo.

Quanto se usam regras para modelagem computacional, é importante entender que não necessariamente uma única regra será suficiente para descrever o comportamento do sistema como um todo. Isto é especialmente verdadeiro quando são descritas situações complexas, tais como eventos biológicos (desenvolvimento celular, redes de regulação genética), previsão de séries e outros.

THOMAS (1998) relata um experimento que descreve uma instância deste problema. Neste experimento foram usadas duas culturas não ativadas de *E. coli* e é adicionada uma grande quantidade de ativador. Imediatamente divide-se a cultura em duas partes (A e B) e dilui-se cada uma das partes de forma que o ativador fica somente em

estado de manutenção (sem ativar nem deixar inibir). A única diferença entre as duas culturas é que na primeira (A) a diluição se deu imediatamente enquanto que na cultura B esperou-se dez minutos antes de se realizar a diluição.

Desta forma, a cultura A que não teve o ativador por tempo suficiente em altas concentrações permanece não induzida enquanto que a subcultura B permanece induzida após ter ficado dez minutos sob atuação do ativador (tempo suficiente para que este realizasse sua "tarefa" de ativação). Logo, a concentração de manutenção do elemento ativador mantém este estado de indução permanentemente.

Algo que é ainda mais interessante é que pode-se misturar as células ativadas e não ativadas no meio com concentração de manutenção de ativador e as células descendentes (as "filhas") estarão no mesmo estado da célula que as geraram (suas células "mães"), não importando o estado original destas. Isto claramente implica que é necessário mais de uma regra para descrever este sistema. A grosso modo, pode-se usar as seguintes:

(a) Se Concentração_Manutenção(Ativador) ^Ativada(Célula) $\Rightarrow$ Ativada(Célula)

(b) Se Concentração_Manutenção(Ativador) ^ ~Ativada(Célula) $\Rightarrow$ ~Ativada(Célula)

Este tipo de regras que aparentemente competem entre si podem ser parte do modelo proposto nesta tese. Se fosse usado um modelo Booleano, o resultado seria que apenas uma das afirmações seria verdadeira, posto que neste modelo, se algo é verdadeiro, sua negação é necessariamente falsa.

Se for utilizado um modelo fuzzy, pode-se ter diferentes graus de pertinência em conjuntos que representam conceitos opostos e, por conseguinte, obter diferentes graus de ativação para cada antecedente, resultando em um modelo relativamente próximo do conceito sendo modelado.

No caso de uma expressão ter pertinência 100% (equivalente ao caso verdadeiro na lógica Booleana) em um conjunto fuzzy, sua negação terá pertinência zero (equivalente ao caso falso). Isto quer dizer que nos extremos a lógica fuzzy se reduz à lógica Booleana tradicional e, por conseguinte, pode-se concluir que a lógica fuzzy não a contradiz, mas sim a complementa, incluindo valores de pertinência intermediários.

Por exemplo, na cultura A, não induzida, a pertinência no conjunto ativada de cada célula será 0 e a sua negação, será 1 (usando-se o operador fuzzy tradicional de negação que consiste na aplicação da fórmula $1 - \mu(x)$). Por conseguinte, se as células forem

colocadas em um meio com uma concentração de manutenção, a pertinência do outro antecedente nas duas regras será um, resultando em uma ativação de zero para a regra (a) e uma ativação de 1 para a regra (b), gerando um próximo estado igual a não ativada para a célula em questão. O mesmo tipo de raciocínio é válido também para a cultura B, invertendo as pertinências para ativada e não ativada.

É importante entender que, posto que é usada a lógica fuzzy, a pertinência em cada um dos conjuntos pode ser diferente de 0 e 1 em qualquer um dos conjuntos em qualquer instante. As regras usadas para modelar o exemplo anterior são essencialmente Booleanas (ou as células estão ativadas ou não o estão), mas, como discutido no capítulo 2 desta tese, a lógica fuzzy permite que se lide com pertinências em vários conjuntos que representam conceitos lingüísticos opostos.

	Número de Conjuntos Fuzzy			
Nomes dos Conjuntos Fuzzy	**1**	**2**	**3**	**5**
	Expresso	Expresso	Pouco Expresso	Muito Pouco Expresso
		Não Expresso	Medianamente Expresso	Pouco Expresso
			Muito Expresso	Medianamente Expresso
				Muito Expresso
				Extremamente Expresso

Tabela 4-1 : Interpretação dos conjuntos fuzzy associados ao conceito de expressão gênica

Assim, além de se usar várias regras diferentes, pode-se assumir que cada elemento sob avaliação está associado com vários conjuntos fuzzy, cuja interpretação é dada na tabela 1 a seguir. Isto difere do modelo discreto (Booleano) no qual cada elemento está ativado (valor 1) ou não ativado (valor 0). Uma interpretação possível para os nomes dos conjuntos fuzzy utilizados, baseando-se no problema de determinação de redes de regulação descrito no capítulo 5 a seguir, é dada na tabela 4-1.

Esta interpretação é baseada no problema – mudando-se o contexto em que se está aplicando o algoritmo, mudam-se os nomes dos conjuntos fuzzy criados. Por exemplo, no caso do problema de classificação para o qual se aplica o algoritmo, conforme descrito no capítulo 6, pode-se interpretar os nomes no caso de dois conjuntos por elemento como representando os conceitos pertence e não pertence.

A opção de se usar um único conjunto foi feita nas aplicações de classificação descritas no capítulo 6 desta tese. Neste caso, precisamos definir se uma tupla pertence ou não a uma determinada categoria, e a interpretação deste conjunto passa a ser "Pertence", cuja pertinência varia de 0 (não pertence) a 1 (pertence).

Os conjuntos fuzzy são criados dividindo-se uniformemente o espaço de expressão do elemento em consideração. O universo de discurso das variáveis fuzzy é delimitado pelos valores mínimo e máximo encontrados nos dados. Por exemplo, sejam os dados exemplo dados pela tabela 4-2.

	Variáveis		
Tupla	**A**	**B**	**C**
1	1,2	2,5	3,4
2	1,4	2,6	3,6
3	1,8	*2,1*	3,2
4	*0,5*	3,0	*2,9*

Tabela 4-2: Dados de um exemplo numérico

Neste caso o valor máximo de cada variável encontra-se sublinhado e o valor mínimo, em itálico. No exemplo, os valores que a variável A pode assumir estão no intervalo [0,5 ; 1,8], que é considerado então o universo de discurso (valores que podem ser assumidos) desta variável. Usando-se a mesma técnica, pode-se concluir que o universo de discurso da variável B é [2,1 ; 2,6] e de C, [2,9 ; 3,6].

Em vários casos pode ser necessário prever valores que estejam fora deste universo de discurso restrito. Isto é especialmente comum em aplicações de previsão nas quais a variável de interesse é uma série que possui uma tendência, seja de crescimento ou de queda. Os valores a serem previstos na hora de aplicação das regra podem então estar fora das faixas de valores encontradas no conjunto de treinamento.

Assim, relaxa-se o universo de discurso, introduzindo-se duas constantes, $\gamma_{min} \leq 1$ e $\gamma_{max} \geq 1$ que multiplicam respectivamente os limites mínimo (l_{min}) e máximo (l_{max}) encontrados nos dados. O universo de discurso considerado então passa a ser então [γ_{min} * l_{min} ; γ_{max} * l_{max}]. Uma divisão prototípica do universo de discurso definido através da aplicação das constantes de relaxamento pode ser vista na figura 4-3.

Uma vez definidos o universo de discurso de uma variável e os conjuntos fuzzy que lhe são associados, resta definir o formato das regras fuzzy utilizadas. Este é similar ao formato normal de regras em RPN descrito anteriormente, de modo que a seguinte gramática descreve uma regra sintaticamente:

<regra> ::= SE <antecedente> ENTÃO <conseqüente>
<conseqüente> ::= <Conjunto_Fuzzy> (<elemento>)
<antecedente> ::= <Conjunto_Fuzzy> (<elemento>) | NOT <antecedente> | AND <antecedente> <antecedente> | OR <antecedente> <antecedente>

Os conjuntos fuzzy têm os nomes descritos na tabela 4-1, notando apenas que foi utilizada uma representação computacional mais eficiente (contendo um número que substitui o nome do conjunto).

Esta descrição é similar à definição BNF descrita anteriormente. Usou-se a mesma estrutura recursiva para o antecedente, que antes se chamava <expressão> e substituiu-se <operandos> por conjuntos fuzzy, que são a base das regras usadas no algoritmo proposto.

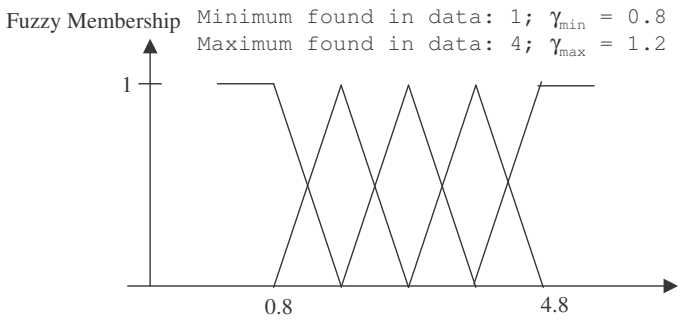


Figura 4-3 → Exemplo genérico de divisão do espaço de expressão em 5 conjuntos fuzzy. Multiplica-se os valores máximos e mínimos pelas constantes pré-definidas resultando em um universo de discurso que é dividido em 4 partes iguais. A metade da primeira parte é atribuída para a primeira função descendente enquanto que a metade da última parte é atribuída para a última função ascendente. As 3 partes restantes são atribuídas para as três funções triangulares restantes. Neste caso imaginário, o mínimo encontrado nos dados foi igual a 1 e a constante γ_{min} é igual a 0,8, o que faz com que o limite inferior do universo de discurso seja igual a $v_{min}*\gamma_{min}$= 0,8. O mesmo raciocínio se aplica para o limite superior do universo de discurso, que é dado por $v_{max}*\gamma_{max}$=4,8.

Como se pode ver pela descrição gramatical acima, foi criada uma expressão usando operadores e os nomes dos conjuntos na representação das regras. Logo, o tamanho da regra não cresce exponencialmente com o número de elementos nela envolvidos, como era o caso da representação binária descrita no capítulo 3. Isto implica em que os cromossomos resultantes não são grandes, sendo computacionalmente para os efeitos de um algoritmo evolucionário.

Cada antecedente de cada regra é uma árvore parecida com aquela vista na figura 4-3. A principal diferença entre a figura e os elementos que formam a regra discutidos agora é que os elementos formadores de uma regra são conjuntos fuzzy ao invés de variáveis, como usado anteriormente. O conseqüente de cada regra é um dos conjuntos fuzzy do elemento em consideração.

Como mencionado anteriormente, é possível haver mais de uma regra por conjunto fuzzy, o que implica em que elas estarão ligadas por um operador OR. Esta possibilidade faz parte do algoritmo evolucionário proposto nesta tese, o que implica em algumas mudanças nos operadores genéticos adotados aqui, como será vista na seção 4.3 deste mesmo capítulo.

O número médio de regras por conjunto fuzzy é um parâmetro do algoritmo. É importante entender que o número de regras para cada um dos conjuntos não necessariamente será igual a este número médio, tendo em vista que a mutação pode

aleatoriamente inserir ou remover regras, alterando o número de regras para cada conjunto. A definição deste operador de mutação é dada detalhadamente na seção c deste mesmo capítulo, mas é importante entender que, graças à sua atuação, os cromossomos têm tamanhos (em número de regras) variados.

Um exemplo da situação descrita no parágrafo anterior pode ser visto na seção 6 do capítulo 5 desta mesma tese, quando é descrito o estudo feito do sistema nervosos central de ratos. Neste exemplo, são definidos 3 conjuntos fuzzy com uma média declarada de 3 regras por conjunto, o que levaria a se esperar que o conjunto de regras obtido tivesse 9 regras no total. Entretanto, o melhor resultado obtido pelo algoritmo proposto possui apenas 7 regras, o que leva a uma média de 2,3 regras por elemento.

Este tipo de variação é esperada em qualquer aplicação prática do algoritmo aqui proposto. Entretanto, para assegurar-se que cada conjunto fuzzy tenha uma participação na obtenção de resultados, qualquer implementação deste algoritmo deve garantir que há pelo menos uma regra associada a cada conjunto fuzzy. Isto permite a obteção de resultados não triviais, pois permite a modelagem de todos os subconjuntos do universo de discurso.

Caso algum conjunto fuzzy não esteja associada a uma regra, ocorrerá uma situação de conjuntos fuzzy sem interseção. Se houver apenas três conjuntos dividindo o universo de discurso e não houver nenhuma regra associada ao conjunto do meio, os dois conjuntos fuzzy vizinhos a este conjunto, que não têm interseção mútua, serão os únicos a afetar o resultado da defuzzyficação. Por conseguinte a única modelagem feita do intervalo representado pelo conjunto sem regras será dada por estes dois conjuntos, isoladamente (cada qual para metade do intervalo).

Esta ausência de interseção faz com que as soluções para este intervalo sejam triviais – dado que o formato da função de pertinência dos conjuntos é triangular, o valor da pertinência obtida multiplicará apenas o máximo do conjunto em que a pertinência não seja zero, gerando uma modelagem equivocada.

Em termos do tamanho do cromossomo, pode-se calcular que se um elemento está asssociado com k conjuntos fuzzy, e o número médio de regras por conjunto fuzzy é m, então haverá um número médio de $k*m$ regras associada a este cromossomo.

Tendo em vista que os operadores de mutação podem alterar este tamanho, eliminando ou acrescentando regras, pode-se chegar a um valor de k regras por

cromossomo, desde que cada conjunto fique associado a apenas uma regra (número mínimo permitido) ou a um valor máximo qualquer, não limitado neste algoritmo. A operação deste operador de mutação quanto à inserção e exclusão de regras é descrita detalhadamente na seção 4.3 deste mesmo capítulo

Qualquer implementação do algoritmo de inicialização da população deve criar aleatoriamente regras respeitando a gramática acima. Entretanto, este tipo de inicialização não é suficiente, pois implicaria em uma busca aleatória, desconsiderando completamente o conhecimento prévio existente. Assim, este algoritmo trabalha com a possibilidade de que o usuário inclua regras que ele considera que devem ser pesquisadas ou obrigatoriamente incluídas/excluídas pelo algoritmo.

Este tipo de conhecimento prévio é muito importante, especialmente na biologia, onde já existem vários relacionamentos previamente comprovados por experimentos e universalmente reconhecidos. Desprezar este conhecimento prévio seria uma estratégia pouco inteligente, a não ser que seja desejado testar o programa para que ele descubra estes relacionamentos bem conhecidos.

O esquema final de definição de um cromossomo é submetido a alguns fatores de influência, dentre os quais podemos ressaltar os seguintes:

- Conjuntos fuzzy, definidos em cima de um universo de discurso que é criado a partir dos valores presentes nos conjuntos dados e de dois parâmetros de relaxamento (γ_{min} e γ_{max}).
- Definição gramatical de uma regra, conforme gramática dada acima.
- Número médio de regras por cromossomo.
- Informação parcial exógena, isto é, informações fornecidas pelo usuário quanto a conhecimento pré-existente na área.
- Atuação de operadores de mutação, que podem alterar o tamanho de um cromossomo, aumentando-o ou diminuindo-o.

A seguir são discutidas certas questões relevantes relacionadas a detalhes práticos da implementação feita no decorrer da elaboração desta tese dos conceitos descritos nesta seção.

4.2.2 Implementação

4.2.2.1 Formato das regras

Na implementação do formato de regras definidas nesta seção buscou-se uma representação que não impingisse nenhum tipo de constrangimento/limitação computacional inapropriado.

Usou-se uma representação interna de árvore, de forma que as estruturas de dados representassem a compreensão mais direta das regras utilizadas.

Ademais, não houve qualquer tipo de restrição aplicável ao gerador de expressões usado para inicializar a população de cromossomos. Este gerador consiste em uma implementação direta da descrição gramatical recursiva citada anteriormente, o que permite que o programa analise toda e qualquer regra que seja gramaticalmente compatível com a descrição em BNF dada acima.

O gerador de funções utiliza um parâmetro denominado coeficiente de altura, determinado pelo usuário. Este coeficiente é usado para determinar se a expressão em questão retornará um operando (conjunto fuzzy) ou uma expressão, continuando a geração de forma recursiva. A decisão é feita através do sorteio de um número no intervalo [0,1] e conseqüente verificação se o número sorteado é menor do que o valor dado por $\frac{1}{2*(coeficiente-1)}$.

Caso o valor seja menor, escolher-se-á através de outro sorteio qual operador será utilizado. Se for escolhido o operador NOT, é necessário gerar mais uma expressão, enquanto que os operadores OR e AND pedem por duas expressões.

A geração destas novas expressões é feita chamando-se a função de inicialização de forma recursiva passando-se como parâmetro o coeficiente de altura usado no sorteio somado de um. Esta soma de uma unidade faz com que seja mais provável a escolha de um

operando na geração da próxima sub-árvore. Isto faz com que seja cada vez mais provável que se gere uma folha conforme a altura da árvore cresce, limitando o tamanho da mesma.

Neste trabalho foram experimentados com vários valores distintos para este coeficiente e descobriu-se que os melhores resultados são obtidos quando o seu valor inicial é definido como 3.

Neste caso, inicialmente haverá $1/4$ de chance de se gerar um operador, e $3/4$ de seleção de um operando, o que faria que a árvore tivesse altura 1. Ao se selecionar um operador, uma nova chamada é feita com o coeficiente com valor igual a 4, o que aufere uma probabilidade de seleção igual a $1/6$ para os operadores de cada sub-árvore necessária, gerando árvores de altura maior ou igual a 3.

O número de árvores que terá altura maior que 3 é menor do que $1/4 * 1/6 * 1/8 \approx 0{,}5\%$, que é a probabilidade de seleção de um operador em cada um dos três níveis mais altos da árvore. Como esta probabilidade é baixa, serão geradas então, em média, árvores baixas que representam regras compreensíveis para o usuário.

Esta restrição quanto à altura média não limita o poder de descoberta do algoritmo, pois este permite que se utilize mais de uma regra para cada conjunto fuzzy conseqüente. O uso de múltiplas regras implica em que as regras para o mesmo conseqüente estarão ligadas por um OR, o que eqüivale a uma árvore de maior altura por conjunto.

4.2.2.2 Embutindo conhecimento prévio

Conforme discutido na seção anterior, o algoritmo aqui proposto utiliza o conhecimento pré-existente para guiar a busca do algoritmo evolucionário, o que permite que sejam buscadas soluções dentro de um espaço restrito, aproveitando-se conhecimento pré-existente.

Criou-se então um mecanismo para permitir que o usuário forneça regras para o programa. As regras entradas são associadas aos conjuntos fuzzy do elemento de interesse e podem ser de três tipos distintos:

- Obrigatórias: o algoritmo garante que elas aparecerão ao menos uma vez nas regras associadas àquele conjunto fuzzy.
- Desejáveis: as regras deste tipo devem preferencialmente aparecer, mas diferentemente das obrigatórias, elas podem ser eliminadas se prejudicarem a avaliação da base de regras. Para que elas apareçam, uma alta percentagem de cromossomos da população inicial (20%) é inicializada com estas regras e depois, quando da aplicação do operador de mutação, é dada a estas regras uma chance de 10% de serem escolhidas. Com estas escolhas, as regras desejáveis têm uma grande chance de serem escolhidas.
- Proibidas: São regras que, como o próprio nome diz, não devem aparecer de nenhuma maneira na base de regras final. Esta flexibilidade é interessante para banir elementos que seriam bons candidatos apenas por uma questão espúria de coincidência dos dados, mas os quais não estão envolvidos no processo regulatório, de acordo com conhecimento obtido *a priori*.

4.3 Função de avaliação

4.3.1 Função utilizada

Para avaliar o desempenho de um cromossomo, são armazenadas t trajetórias com ns_t passos cada. Cada trajetória representa o comportamento real da rede de regulação que se deseja modelar, isto é, os valores que cada uma das variáveis do conjuntos de dados assume em cada instante t. A partir deste ponto em diante será usado o adjetivo “real” para descrever cada trajetória da rede sendo modelada.

O objetivo da função de avaliação é determinar quão bem o cromossomo modela o comportamento real dos dados. Isto é feito determinando se as regras codificadas em um cromossomo fazem com que a trajetória real seja replicada de forma precisa. A trajetória real consiste, no caso da aplicação de microarrays, das medidas consecutivas feitas sobre o mesmo conjunto de genes.

Para tanto, é necessário determinar o valor de cada uma das variáveis em cada instante de tempo, valores estes que são calculados aplicando-se as regras fuzzy codificadas no cromossomo.

Assim, para verificar quão bem cada cromossomo modela uma trajetória, é necessário verificar se cada um dos estados (instantes t=1,..,k) da trajetória é corretamente previsto, tendo como base somente o seu estado anterior (instante t-1).

Assim, para avaliar cada cromossomo usa-se o primeiro estado de cada trajetória (valor de todas as variáveis no instante t=0) para calcular o valor destas variáveis no instante t=1. Este cálculo é realizado usando-se a base de regras fuzzy para iterar os valores.

Este processo é repetido para os instantes t=2,...,k, até chegar no instante final da trajetória. Para cada um destes instantes calcula-se o seu estado usando-se o passo do instante *t-1* como base para calcular o valor de todas as variáveis. Estes valores calculados através da aplicação das regras fuzzy serão denominados doravante de trajetória calculada.

No capítulo anterior, ao lidar com dados Booleanos, determinou-se a função de avaliação somando-se o número de bits da trajetória calculada que diferem dos bits da trajetória real e dividindo-se o valor obtido por uma média pelo número de passos. Isto pode ser feito pois no caso de redes Booleanas não há outro tipo de erro que não o de valor absoluto, pois ou o ponto previsto está correto (possui valor igual ao desejado, que é aquele originalmente na trajetória) ou não está (possui valor diferente do desejado), logo uma contagem simples dos valores errados basta para aferir a qualidade da previsão (LINDEN *et al.* 2002a, LINDEN *et al.*, 2002b).

No caso contínuo é diferente, pois os erros não são absolutos. Para cada um dos valores calculados existe uma diferença finita entre os valores previsto e o real, valor este que tem um significado diferente de elemento para elemento.

Esta diferença de significado depende do valor absoluto do elemento, o que implica no fato de uma diferença percentual ser mais significativa que a diferença absoluta. Por exemplo, se o valor absoluto é 0,1, uma diferença entre o valor previsto e o real de 0,02 é bastante significativa (20% de erro), enquanto que se o valor original da variável é 25, o mesmo erro absoluto corresponde a um erro percentual de aproximadamente 0,1%, sendo praticamente desprezível na maioria das aplicações

Assim, ao invés de usar uma diferença absoluta, é usado o erro percentual médio absoluto (mean absolute percentage error ou MAPE) como medida de qualidade de previsões. Este erro é dado pela seguinte fórmula:

$$MAPE = \frac{1}{N}\sum_{t=1}^{N}\left|\frac{y_t - \hat{y}_t}{y_t}\right| \quad (1)$$

Este valor é então dividido pelo número de passos por trajetória para minimizar o efeito das diferenças de tamanho entre trajetórias, como descrito no caso Booleano. É importante compreender que se todas as trajetórias tiverem o mesmo tamanho ou se houver apenas uma única trajetória, esta divisão não afeta o resultado final. Uma vez calculado o valor do erro médio por trajetória, somam-se os erros de todas as trajetórias, obtendo uma métrica que é dada pela seguinte fórmula:

$$\frac{\sum_{t=1}^{\#Traj}\left(\frac{\sum_{g=1}^{\#genes} MAPE_g}{\#Steps_t}\right)}{\#Traj} \quad (2)$$

Este valor é então invertido e resulta em um número adimensional que não tem nenhum relacionamento óbvio com qualquer medida de erro. Para que o significado deste número seja melhor compreendido, é calculado um valor denominado de “infinito”, que corresponde ao valor ideal previsto pelo algoritmo.

Tendo em vista que o erro médio é invertido na fórmula (2), não se pode usar este valor como nível de perfeição. Ademais, na maioria das aplicações práticas o erro zero é

um valor inatingível, podendo-se estabelecer um patamar de erro máximo aceitável, para o qual se considera que o algoritmo foi bem sucedido em sua previsão.

Assim, calcula-se o valor do "infinito" através da aplicação na fórmula (2) de um percentual de erro determinado pelo usuário em cada elemento em cada passo da trajetória. Por exemplo, um nível de erro de 0,5% em uma única trajetória gera uma medida de qualidade de valor 200. Usando este nível como padrão-ouro, podemos avaliar a qualidade das soluções obtidas de forma utilizável em uma estratégia de roleta viciada em um algoritmo evolucionário.

Dado que não é usado um cromossomo representado por uma *string* de bits, é necessário criar uma nova maneira de se recompensar minimalidade. Isto decorre do fato anteriormente mencionado de que se dois cromossomos mapeam uma trajetória com a mesma qualidade, o menor e mais simples deve ser preferido.

Esta preferência pela minimalidade é conseqüência natural do fato de que os resultados deste programa serão interpretados por pesquisadores e soluções mais simples e elegantes tendem a ser mais apreciadas. Ademais, quando se discute aplicações biológicas da natureza, é interessante modelar o fato de que esta costuma recompensar minimalidade, já que redes com menos caminhos regulatórios tendem a ser mais dificilmente rompidas por eventos aleatórios tais como mutações, isto é, são mais robustas.

No caso da modelagem contínua, foi criado então um coeficiente de penalização de soluções longas. Este coeficiente $c \leq 1$ é multiplicado pela função de avaliação do cromossomo e é dependente da altura média de todas as árvore de regras no cromossomo em avaliação (h) e do número médio de regras por variável naquele cromossomo (n), sendo dado então pela seguinte fórmula:

$$\begin{cases} c = \dfrac{1}{n}, h \leq 2 \\ c = \dfrac{1}{(h-1)*n}, h \geq 2 \end{cases} \tag{3}$$

Quanto maior a altura da árvore ou o número médio de regras por elemento, maior a penalização aplicada sobre o cromossomo quando do cálculo da função de avaliação. Esta penalização faz com que um cromossomo mais curto e mais simples passe a ser preferido

sobre outro maior que comete o mesmo erro no cálculo das trajetórias, pois a avaliação do cromossomo mais curto passa a ser mais alta do que a do mais longo. O cromossomo mais curto de avaliação inferior pode passar a ser preferido em detrimento do cromossomo maior de melhor avaliação, desde que a diferença entre os coeficientes de preenchimento seja maior do que a diferença entre as avaliações.

É importante entender que quando usamos o coeficiente de preenchimento, em muitos casos o valor calculado de "infinito" nunca será atingido pelos cromossomos da população sendo evoluída, mesmo que o nível de erro usado para calcular o valor do infinito tenha sido atingido por um cromossomo candidato. Por exemplo, se um cromossomo com um número médio de regras igual a 2 e árvores de altura média igual a 2 atingir o nível de erro desejável, a sua função de avaliação será igual, pela aplicação da fórmula (3), a *infinito/2*. Isto não é problemático, sendo apenas uma questão de interpretação dos resultados dados. Ao avaliar os resultados dados pelo programa, é importante que se entenda que o valor de "infinito" só será atingido se uma rede obtiver o erro desejado com uma regra por elemento sendo que cada regra possui altura igual a 2 ou menos.

O uso deste coeficiente é algo interessante até quando se pensa na qualidade dos resultados obtidos. VAN SOMEREN *et al.* (2000) provam que o número de pontos no tempo necessários para identificar corretamente os pesos de uma rede esparsamente conectada, como aquelas geradas por este algoritmo, é da ordem da conectividade máxima da rede.

Logo, cromossomos que possuem regras mais curtas e que representam redes de regulação menos conectadas fornecem resultados mais significativos com a mesma quantidade de dados.

4.3.2 Avaliações separadas

SUER *et al.* (2002), introduzem a idéia de que quando forem buscadas a regulação de múltiplos elementos simultaneamente, pode-se avaliar cada parte do cromossomo

separadamente (isto é, realizar todo o processo do algoritmo evolucionário por nó, ao invés de por cromossomo completo). Isto pode ser interessante posto que um cromossomo pode ser bom para regular um nó enquanto que ele é ruim para regular outros, como pode ser visto no exemplo sintético mostrado na tabela 4-3.

Trajetória Real			**Trajetória Calculada**		
Gen1	*Gen2*	*Gen3*	*Gen1*	*Gen2*	*Gen3*
1,5	2,5	4,2	1,5	2,5	4,2
2,3	3,7	3,4	2,4	3,9	3,4
3,2	2,4	2,9	3,15	4,2	1,5
4,3	1,2	2,1	4,39	4,3	1,2
Erro por elemento:			**0,026**	1,129	0,303
Erro Total:			**0,486**		

Tabela 4-3: Exemplo de uma regulação adequada para um único elemento, mas que não consegue capturar todas as regulações possíveis. No caso do Gen1, o erro calculado é baixo, indicando boa adequação do cromossomo. No caso dos outros dois genes, o erro é muito alto, levando o cromossomo a ter uma avaliação total ruim, posto que esta é baseada na média dos erros.

Como pode-se ver na tabela 4.3, o erro cometido para o nó Gen1 (em negrito) é baixo. Ademais, as tendências para este nó são captadas de forma precisa pela sua estratégia de controle. Entretanto, a avaliação para os outros dois nós (Gen2 e Gen3) é tão ruim que faz com que o cromossomo com um todo tenha uma péssima avaliação geral, dando a ele uma baixa chance de ser escolhido.

Considerou-se então modificar o algoritmo evolucionário proposto aqui de forma a selecionar diferentes pais para cada um dos nós, de forma que o operador de crossover pudesse envolver múltiplos cromossomos (dois para cada nó), ao invés da abordagem clássica de escolher apenas um par de cromossomos que intercambiam todo o seu material genético.

O uso desta estratégia teoricamente permite que sejam usados cromossomos que são eficientes em cada um dos nós gerando soluções com qualidade ainda melhor do que as atualmente geradas.

Entretanto, esta estratégia é similar a se realizar múltiplas execuções do algoritmo, pelos seguintes motivos:

- É necessário manter-se populações independentes para cada um dos nós cuja regulação está sendo buscada.
- As avaliações para cada nó são feitas de forma independente
- O número de aplicações dos operadores genéticos é equivalente ao número que seria feito em execuções independentes. Como cada nó é avaliado de forma independente, cada um deles precisa ter uma roleta separada com a conseqüente seleção de pais distintos e aplicação dos operadores também de forma separada.

Todos estes fatores levam à conclusão de que não há ganhos reais quanto a esta estratégia, tendo em vista que na definição de (SUER *et al.*, 2002) não existe uma interdependência entre as regulações de forma que haja vantagens definitivas em realizar uma única execução ao invés de múltiplas.

Cabe ressaltar que quando um elemento não está sendo avaliado durante a execução do algoritmo proposto nesta tese, ele tem os seus valores reais usados em cada um dos instantes de tempo. Assim, pode-se realizar múltiplas execuções para buscar estas regulações, sem necessidade de aplicar o conceito de avaliações em separado que não proporcionaria ganhos reais para os resultados obtidos.

Com isto, esta estratégia foi abandonada, não tendo sido incluída na versão final da ferramenta desenvolvida para esta tese.

4.4 Operadores Genéticos

Para completar a definição de algoritmo evolucionário (EA) proposto, é necessário definir os operadores de mutação e de recombinação (crossover) usados. Como de praxe em grande parte da literatura, usam-se dois operadores que competem entre si.

O EA definido nesta tese alterna entre os operadores de mutação e crossover usando uma roleta viciada baseada em probabilidades fixas ou que variam com o tempo, conforme o valor de um parâmetro do algoritmo.

Como será visto nos experimentos relatados nos capítulos 5 e 6, a variação linear da fitness dos operadores foi a forma usada. Isto decorre do fato de que no começo é desejável recombinar as soluções de forma a utilizar o máximo possível de material genético (esquemas). Entretanto, quando a execução se aproxima do fim ocorre o efeito da convergência genética que faz com que a população seja muito similar e o uso mais freqüente da mutação faz mais sentido. Uma aplicação desta variação das probabilidades associadas aos operadores se encontra na seção 6 do capítulo 5 desta mesma tese.

4.4.1 Operador de crossover

O operador de crossover tem como objetivo realizar uma troca de informação entre dois indivíduos da população de uma maneira análoga à reprodução sexuada. Seu uso implica no intercâmbio entre dois indivíduos de pedaços de antecedentes de regras ("material genético"), gerando dois "filhos" que possuem fragmentos de regras de cada um dos pais, compartilhando suas qualidades na modelagem dos dados.

Este processo é análogo ao processo de recombinação gênica (crossing-over) que acontece na natureza, onde dois alelos são recombinados através da troca mútua de segmentos entre pares de cromossomos (HOLLAND, 1975).

O operador de crossover do algoritmo proposto nesta tese funciona em dois níveis: o primeiro é o intercâmbio de material entre duas regras com o mesmo conseqüente, enquanto que o segundo é o controle em nível de cromossomo.

O primeiro nível consiste no intercâmbio de sub-árvores entre duas regras que têm o mesmo conseqüente. Por exemplo, a regra *SE A AND (B OR NOT C AND D) ENTÃO F(X)* e a regra *SE NOT B OR (C AND NOT D) ENTÃO F(X)*. Escolhe-se então um nó aleatoriamente em cada uma das árvores e realiza-se o intercâmbio entre as sub-árvores

enraizadas em cada um destes nós. Um exemplo da operação do crossover pode ser visto na figura 4-4.

O segundo nível do operador consiste em um controle da seleção das árvores a serem usadas pelo primeiro nível e é necessário pois pode-se ter múltiplas regras por conjunto fuzzy, o que implica em duas coisas:

- É necessário escolher com qual das regras do outro conjunto a regra atual irá realizar o intercâmbio genético.
- É necessário garantir que a regra escolhida contenha o mesmo conseqüente que a regra corrente.

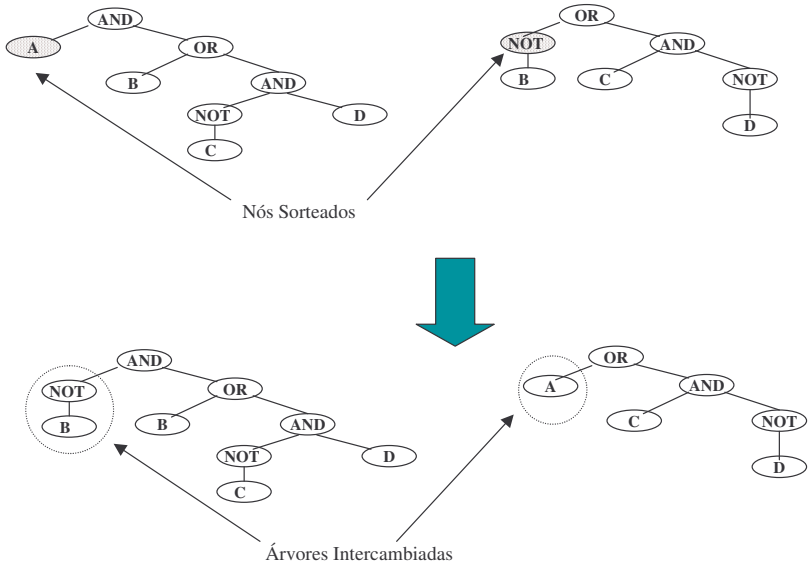


Figura 4-4 → Exemplo de aplicação do operador de crossover. Sorteia-se um nó de cada sub-árvore (o nó A na árvore da esquerda e o nó NOT na sub-árvore à direita, marcados com fundo cinza) e intercambiam-se as sub-árvores enraizadas em cada um deles. O resultado é visível na parte de baixo da figura.

Assim, qualquer implementação deste algoritmo precisa garantir que as regras para o conjunto X do cromossomo 1 (c_1) somente realizam o crossover ("cruzam") com as regras para o mesmo conjunto fuzzy no cromossomo 2 (c_2), mesmo que um dos cromossomos tenha mais regras para este conjunto do que o outro. Isto quer dizer que se o cromossomo c_1 possui duas regras para o conjunto X e o cromossomo c_2 só possui uma, as duas regras de c_1 realizarão o intercâmbio genético descrito acima com a mesma regra de c_2.

Se a situação for inversa, e o cromossomo c_1 tem apenas uma regra enquanto que c_2 tem duas, pode-se escolher de forma aleatória qual das regras de c_2 irá cruzar com a regra de c_1 e o filho resultante terá apenas uma regra para este conjunto fuzzy. Como de praxe nos algoritmos evolucionários, o processo de crossover ocorre duas vezes gerando dois filhos, logo uma implementação possível do algoritmo é realizar o mesmo processo duas vezes, uma varrendo cada pai, logo na segunda execução, o papel de c_1 e c_2 estarão invertidos. Exemplos de ambas as situações são mostrados na figura 4-5 (a).

Se ambos os cromossomos tiverem mais de uma regra para o conjunto fuzzy cujas regras estão sendo consideradas para o crossover, é interessante que qualquer implementação do algoritmo garanta que cada regra realizará ao menos um cruzamento.

Tendo em vista que a lógica fuzzy usa todas as regras da base de regras para chegar a uma conclusão, é interessante garantir que as características de todas as regras estejam presentes na descendência dos cromossomos. Isto é, cada regra tem uma contribuição específica, logo, é interessante que seus antecedentes sejam considerados para a geração dos novos indivíduos, de forma que esta contribuição seja preservada.

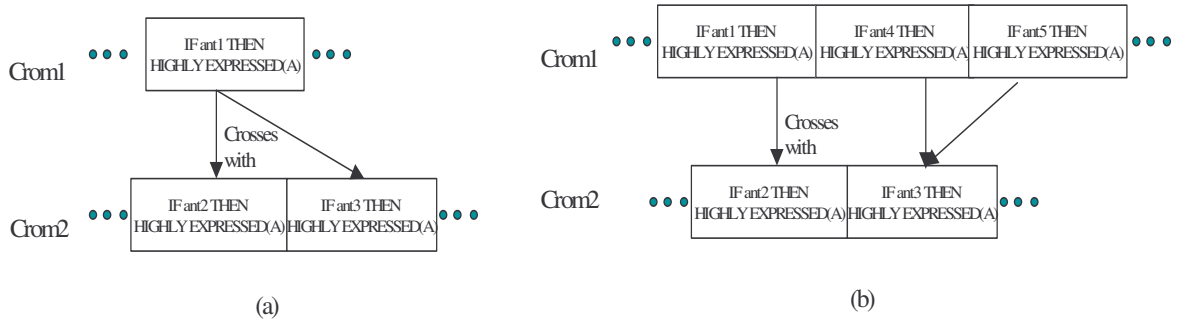


Figura 4-5 → Exemplo de execução do controle em nível de regras do operador de crossover. Em (a) pode-se ver a situação em que o cromossomo 1 tem apenas uma regra para o conjunto sob análise. Neste caso, esta única regra realiza o crossover com as duas regras do crmossomo 2. Em (b) pode-se ver a situação em que ambos os cromossomos têm mais de uma regra para o conjunto fuzzy. Qualque rimplementação pode escolher quais regras cruzarão de forma aleatória, mas deve garantir que cada regra realize o crossover ao menos uma vez.

4.4.2 Operador de mutação

O operador de mutação tem como função inserir variabilidade genética na população sendo evoluída. Sem a aplicação deste operador, a população tende a convergir em algumas gerações para um grupo muito pequeno de cromossomos. O tempo de convergência é função apenas do tamanho do espaço de busca e do tamanho da população, mas esta não deixa de ocorrer.

No caso do algoritmo evolucionário aqui proposto, são aplicados três tipos distintos de operadores de mutação:

- Operador de mutação de regras
- Operador de inserção de regras
- Operador de exclusão de regras

O operador de mutação de regras escolhe um nó aleatoriamente em uma árvore de regra e elimina toda a sub-árvore enraizada naquele nó. Posteriormente, uma nova sub-árvore é gerada da mesma maneira que os cromossomos da população inicial. O *modus operandi* deste operador de mutação é descrito na figura 4-6

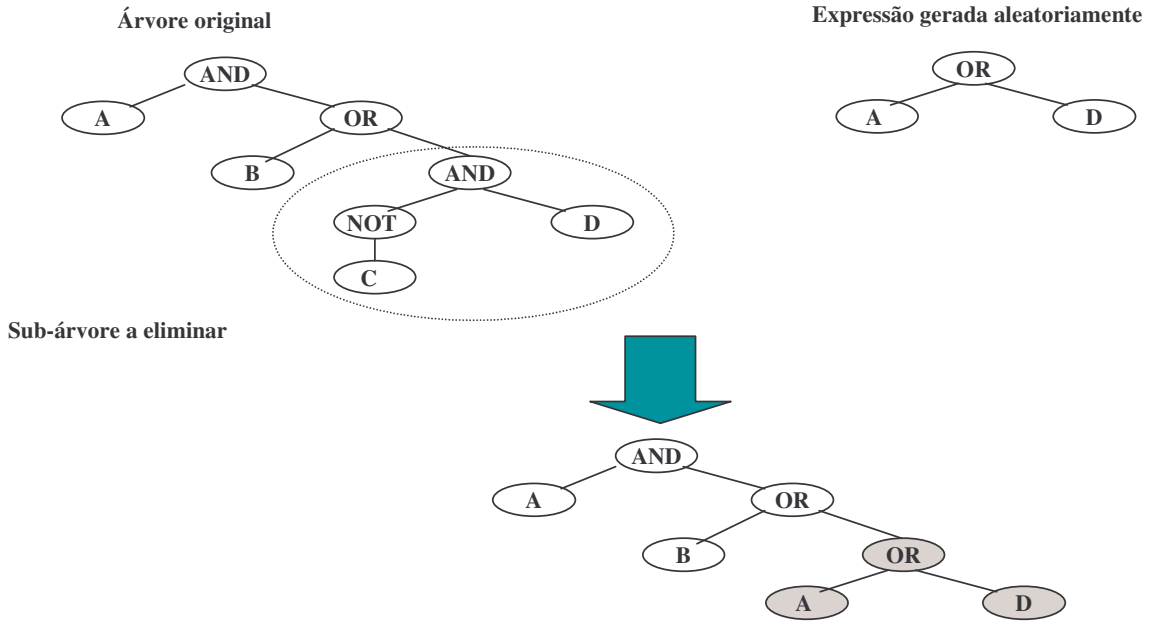


Figura 4-6 → Exemplo de operação do operador de mutação de regras. Um nó é escolhido de forma aleatória (no caso do exemplo, o nó AND) e a sub-árvore enraizada nele (circulada) é substituída por uma expressão gerada de forma aleatória.

O operador de inserção de regras escolhe um conjunto fuzzy aleatoriamente e cria uma nova regra para ele, usando o inicializador da população, enquanto que o operador de exclusão de regras elimina uma regra para um conjunto escolhido aleatoriamente, desde que este conjunto tenha mais de uma regra.

Esta restrição decorre do fato de que é necessário garantir que todos os conjuntos fuzzy tenham pelo menos uma regra associada, para que possam contribuir para a função de avaliação.

Estes últimos dois operadores (inclusão e exclusão de regras) alteram o número de regras por conjunto fuzzy, fazendo com que seja possível que um gene específico tenha um número de regras diferente do número médio definido como parâmetro do algoritmo.

Se a implementação garantir que ambos possuam a mesma probabilidade, então a média global do número de regras não será afetada, pois alguns genes terão seu número aumentado enquanto que para outros este número diminuirá, e o número global de regras tenderá a permanecer estável.

4.4.3 Implementação dos operadores

Há vários detalhes dos operadores que devem ser levados em consideração de forma cuidadosa quando for feita uma implementação do algoritmo. Nesta seção listamos os principais fatores.

Quando se está aplicando o crossover entre duas regras, varre-se uma regra de cada vez. Em cada sub-árvore da árvore sendo visitada é decidido de forma aleatória se será feito uma troca ou não. Se o sorteio retorna positivo, é feita uma troca entre a sub-árvore corrente completa (isto é, a sub-árvore enraizada no nó corrente que estamos visitando) e a sub-árvore equivalente do outro pai. Um exemplo desta varredura é mostrado na figura 4-7, que consiste em uma representação algorítmica do processo mostrado em linhas gerais na figura 4-4 acima.

O crossover tradicionalmente gera dois filhos por operação. Para fazê-lo, a implementação aplica o algoritmo descrito na seção anterior duas vezes: primeiro considerando os pais na ordem c_1/c_2 e depois considerando os mesmos pais na ordem c_2/c_1.

A escolha das regras que irão cruzar é feita de forma aleatória, dada a existência de várias regras alternativas para um mesmo conjunto. A implementação garante que cada regra realize ao menos um cruzamento de forma a preservar sua contribuição para a qualidade geral do cromossomo, seguindo o esquema mostrado na figura 4-5 (b).

A implementação do operador de mutação necessita que sejam criadas novas sub-árvores, mas se estas forem criadas da mesma forma que a população original, a tendência é que a altura média das árvores sofrendo mutação cresça.

Para evitar este fenômeno, que comprometeria a simplicidade das regras, o módulo inicializador é instruído para gerar árvores baixas, com altura (maior número de arestas entre a raiz e as folhas) menor ou igual a 3, o que é feito acertando o coeficiente de altura, como discutido no item b.2) desta mesma seção.

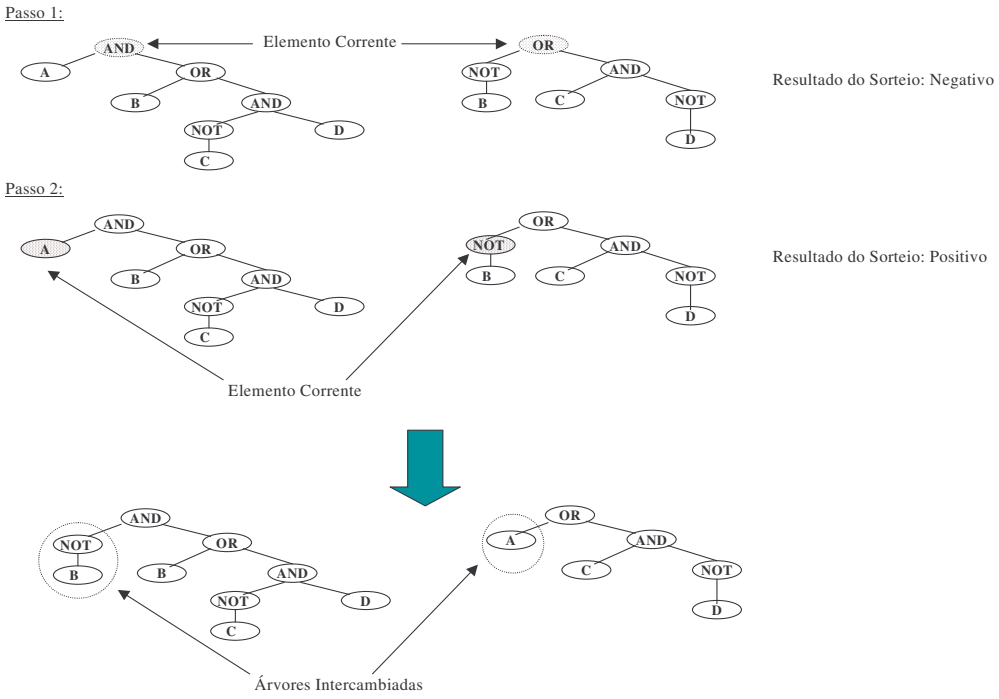


Figura 4-7 → Exemplo da operação de crossover em nossas árrvores

4.5 Critérios de parada

No EA proposto nesta tese foram usados três diferentes critérios de parada:

- Baseado no número de gerações (não mais do que 80 gerações por rodada)
- Baseado no conceito de estagnação devido à convergência genética e/ou fatores estocásticos, isto é, interromper a execução se a melhor solução não melhorou durante as últimas 20 gerações
- Baseado na qualidade da solução, o qual faz com que a execução seja interrompida caso o erro máximo cometido nas previsões seja inferior a um padrão determinado pelo usuário

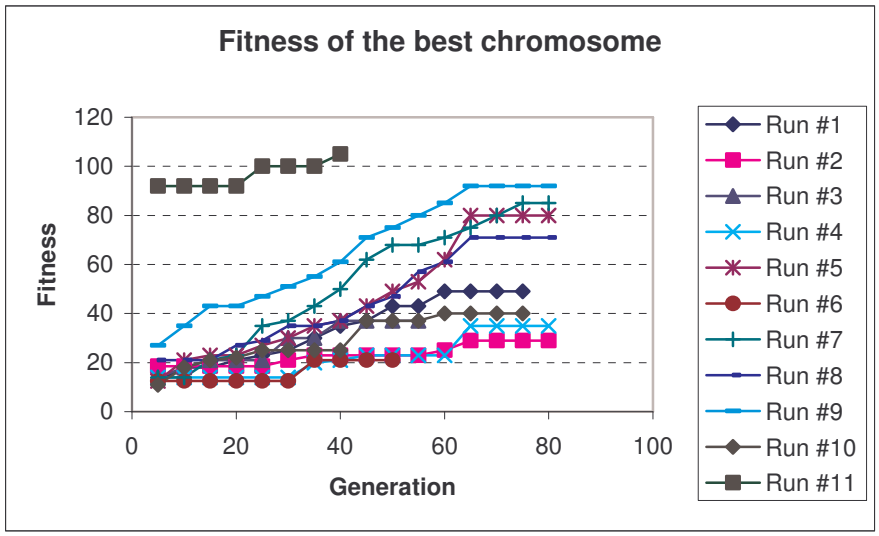


Fig. 4-8: A evolução da avaliação do melhor cromossomo durante a descoberta da regulação para o gene preGAD67. A avaliação foi calculada como descrita na seção 4b) e o número 100 corresponde a aproximadamente 1% de erro na previsão do valor de expressão em cada instante de tempo. Varias curvas páram antes da octogésima geração devido à aplicação de outros critérios de parada. As curvas relativas às execuções de número 1 e 10 param na geração 70 devido à estagnação da avaliação do melhor indivíduo da população. A curva correspondente à 11ª. execução, por exemplo, pára na geração 40 devido à precisão da solução obtida.

O limitador do número de gerações foi determinado empiricamente, tendo sido visto em várias execuções que ele é superior ao número de gerações em que se percebe uma convergência genética da população. Este efeito pode ser visto na figura 4-8, a qual

descreve 11 execuções feitas para uma das aplicações explicadas no próximo capítulo (a determinação de um conjunto de regras para o gene preGAD67, um dos genes relevantes para o desenvolvimento do sistema nervoso central de ratos).

4.6 Dados categóricos

Ao procurar dados para testar o algoritmo proposto nesta tese, verifica-se que nem sempre os conjuntos disponíveis possuíam apenas atributos numéricos. Na verdade, na maioria das vezes, os conjuntos de dados disponíveis incluíam dados categóricos.

Dados categóricos são tipos de variáveis que consistem em uma série de estados discretos tal que cada objeto pertence a apenas um dos estados. Exemplos são a patente de um oficial do exército, a cor de seus olhos, o aminoácido em uma determinada posição de uma proteína. Estes dados são chamados de nominais quando não existe uma ordem específica entre os estados diferentes (por exemplo, entre olhos azuis e castanhos) e de ordinais quando existe uma ordem subjacente que pode ser imposta aos dados (por exemplo capitão>tentente>sargento) (GORDON, 1981).

No caso dos dados nominais, não se pode estabelecer uma função de distância natural entre dois elementos. GIBSON *et al.* (2000) apresenta um exemplo que consiste em analisar um banco de dado de informação de carros, contendo fabricante, modelo, preço, cor e cliente. O fabricante é um típico exemplo de dado categórico nominal: é difícil dizer qual dado é mais ou menos próximo a outro. Por exemplo, “Honda” e “Toyota” podem ser considerados próximos se pensarmos na origem da fábrica, mas talvez nem tanto se pensarmos na faixa de preço média de seus modelos. Desta forma, não se pode usar métricas tradicionais, tais como distância euclidiana ou outras, sendo necessária outra forma de lidar com estes dados, como será visto em uma seção mais adiante.

Para poder tratar este tipo de dados foram introduzidas algumas modificações na estrutura dos cromossomos usados no algoritmo aqui proposto, modificações estas que serão descritas na próxima seção.

4.7 Modificações no cromossomo

A estrutura do cromossomo usado pelo algoritmo aqui proposto precisou ser modificado para poder lidar com dados categóricos. Este tipo de dado não se presta a uma abordagem fuzzy, tendo em vista que não são ordenáveis, não sendo possível então criar conjuntos fuzzy para definir uma função de pertinência.

Assim, foi necessário capacitar nossos cromossomos para lidar com conjuntos crisp (não-fuzzy) ao mesmo tempo em que lidavam com conjuntos fuzzy. A sintaxe de nossas pré-condições passou então a ser dada pela mesma gramática que usamos anteriormente na modelagem contínua:

<expressão> ::= AND <expressão> <expressão>| OR <expressão> <expressão>|NOT <expressão> | <operando>

Entretanto, neste tipo de operação <operando> é definido de forma diferente:

<operando> ::= <conjunto_fuzzy>(<atributo>) | <atributo> IN { <valores> }*

,onde <valores> corresponde a um conjunto contendo zero ou mais dos valores que existem na coluna do <atributo> dentro do banco de dados. Para entender como inicializamos os conjuntos de atributos, veja a seção e) deste mesmo capítulo.

4.8 Inicializando os conjuntos de valores categóricos

Como já discutido em uma seção anterior deste mesmo capítulo, muitos conjuntos de dados incluem variáveis categóricas, isto é, variáveis não numéricas que podem conter

informações relevantes para auxiliar na classificação dos dados em questão. Nós modificamos então a estrutura do nosso cromossomo de forma a permitir que este tipo de variável participe de nossas regras, conforme descrito na seção (c) deste mesmo capítulo.

Esta modificação implica em um tratamento de conjuntos não fuzzy, que consistem em uma partição crisp de nossos dados. Entretanto, é necessário descobrir a partição dos dados em conjuntos.

É possível modificar o GA de forma que este tente descobrir sozinho os conjuntos de elementos categóricos a serem colocados nas nossas regras, mas existem técnicas mais eficiente de fazê-lo. Estas incluem a utilização de um algoritmo de agrupamento de dados categóricos de forma a gerar conjuntos que tivessem um significado extraído da estrutura subjacente aos dados.

GIBSON *et al.* (2000) descreve o algoritmo usado para fazer esta análise (STIRR), algoritmo este que usa uma técnica baseada em sistemas dinâmicos que itera uma série de pesos entre valores armazenados de forma a criar grupos de elementos com alto grau de co-ocorrência e/ou associação.

Basicamente, o algoritmo funciona inicializando um grafo cujas arestas consistem nos valores que co-ocorrem em uma mesma tupla. É óbvio que elementos do mesmo atributo (mesma coluna) nunca estarão ligados, pois não temos múltiplos valores por atributo em um banco de dados, mas é possível que haja um caminho entre eles, dependendo dos elementos de outras colunas que co-ocorrem com os dois.

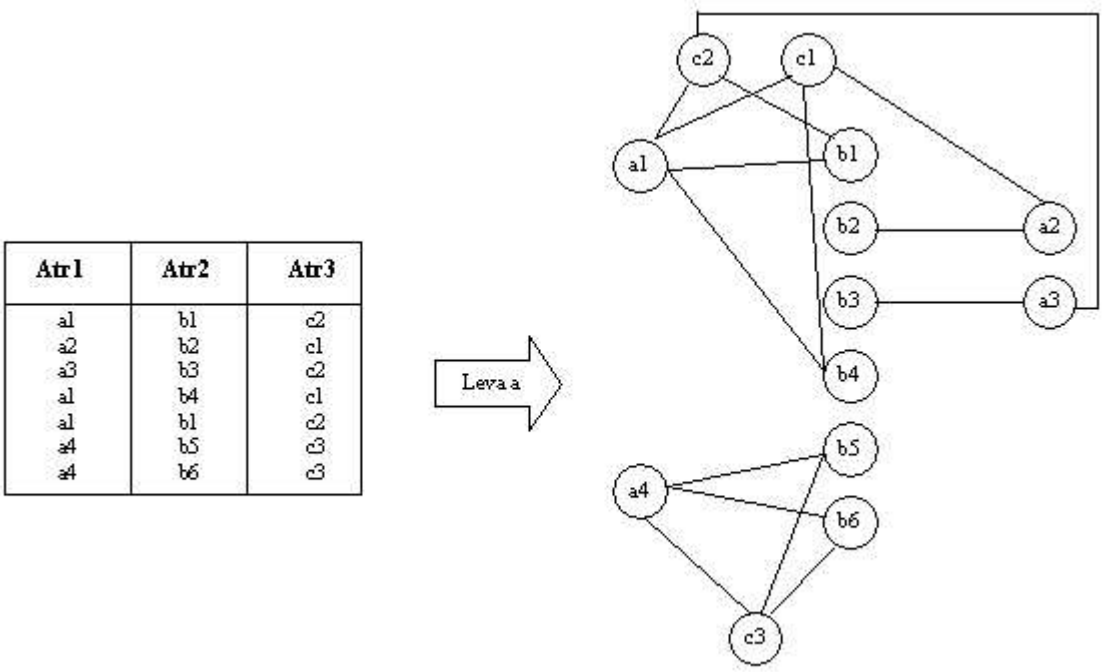


Figura 4-9 → Exemplo de inicialização do grafo usado no algoritmo utilizado. Na primeira linha da base de dados exemplo, pode-se ver que a1 co-ocorre com b1 e c2. Logo, criamos arestas entre os nós que representam estes valores. Como a2 e a1 co-ocorrem com c1, existe um caminho entre os dois, o que pode indicar uma associação, mesmo que seja impossível para os dois valores co-ocorrerem na mesma tupla.

A nossa maneira de inicializar o grafo é ligeiramente diferente daquela sugerida nas figuras e exemplos do artigo em questão. Nós procuramos inserir uma ligação entre todos os valores que co-ocorrem em uma tupla, de todas as colunas para todas as colunas, enquanto que os exemplos do artigo supracitado fazem uma ligação de uma coluna para a outra. Esta alteração reforça as ligações entre os elementos que devem estar no mesmo cluster, pois cria mais caminhos de um nó contendo um elemento para outro de uma mesma coluna que tenha uma forte associação com o primeiro. Um exemplo desta inicialização pode ser visto na figura 4-9.

Uma vez montado o grafo, atribuímos um peso a cada um dos nós, a guisa de inicialização do sistema. Estas arestas podem ser inicializadas das seguintes formas:

- Aleatória → todos os nós recebendo valores entre zero e um escolhidos de forma aleatória.
- Uniforme → todos os nós recebem um valor igual a um.

- Sobre um nó → escolhemos um valor de interesse e todos os valores que co-ocorrerem em tuplas com este elemento serão inicializados com valor 1. Todos os outros nós serão inicializados com valor zero.

Após a inicialização, todos os valores são normalizados para evitar saturação das funções de iteração. No nosso caso, nós inicializamos o sistema sobre cada um dos nós e extraímos os elementos pertencentes ao grupo deste elemento, conforme descreveremos mais à frente neste mesmo capítulo.

Uma vez inicializado o sistema, nós o iteramos usando uma função de combinação *f*. Para cada tupla τ contendo um elemento de interesse v e os nós u_1, u_2, ... u_{n-1}, nós calculamos o valor x_τ dado por $f(u_1, u_2, ... u_{n-1})$. O peso atribuído ao nó v será então dado por $\sum_\tau x_\tau$.

Após os cálculos, os pesos são novamente normalizados. O processo de iteração é repetido então até que o sistema atinja um estado estável, configuração esta denominada de bacia de atração do sistema.

Entre as possíveis funções de combinação de elementos, nós temos as seguintes:

- Produtório: $\Pi(u_1, u_2, ... u_{n-1}) = u_1 u_2 ... u_{n-1}$
- Somatório: $\Sigma(u_1, u_2, ... u_{n-1}) = u_1 + u_2 ... + u_{n-1}$
- Operador somatório generalizado: $S_p(u_1, u_2, ... u_{n-1}) = (u_1^p + u_2^p ... + u_{n-1}^p)^{1/p}$

Nas execuções de nosso sistema nós percebemos que o operador produtório é sensível a valores próximos de zero, fazendo o sistema convergir de forma extremamente rápida para uma configuração em que a grande maioria dos pesos é igual a zero.

Já o operador somatório generalizado tende a fazer com que o maior valor dentre os pesos predomine sobre todos os demais, negligenciando a contribuição de todos os outros elementos. Isto é um fato normalmente verificado em situação de agrupamento em que se utiliza métricas de Minkowski.

Desta forma nós selecionamos o operador somatório, que é aquele que produz menos efeitos espúrios e leva em consideração todos os pesos de todos os nós com arestas ligando-os ao nó em consideração.

GIBSON *et al.* (2000) oferece provas da convergência do algoritmo, com a característica especial de que quase todas as configurações iniciais do sistema convergirão para o mesmo ponto, que consiste no autovetor principal do sistema, visto como uma configuração.

Considerando que usamos o somatório como operador de atualização de nossos pesos, e que usamos uma função N de normalização global a cada iteração, o algoritmo de iteração dos pesos w_x pode ser colocado da seguinte forma:

Inicialize a configuração W.
Enquanto não convergir, faça
    ∀nó u_i ∈ grafo faça
        ∀tupla τ { $u_{\tau 1}$, $u_{\tau 2}$, ... , $u_{\tau i}$, ..., $u_{\tau n}$ } contendo u_i faça

$$x_\tau \leftarrow w_{u_{\tau 1}} + w_{u_{\tau 2}} + ... + w_{u_{\tau i}} + ... + w_{u_{\tau n}}$$

$$w'_{u_i} \leftarrow \sum_\tau x_\tau$$

    W ← N (W′), onde N é a função de normalização global
Fim Enquanto

ZHANG *et al.* (2000) tentam demonstra que as provas oferecidas pelo artigo em questão são insuficientes e que existem alguns estados iniciais especiais que podem levar a uma não convergência do algoritmo. Eles propõem duas alterações fundamentais no algoritmo aplicado: alterar a matriz de iteração do sistema, de forma a multiplicar o peso associado ao elemento corrente por uma constante positiva maior que um e utilizar uma função de normalização global, isto é, fazer com que a soma de todos os pesos seja um, e não a soma de cada coluna.

O problema apontado pelos autores consiste em uma má formação da matriz de iteração do método STIRR. Este realiza uma projeção dos dados no espaço do autovetor dominante. Entretanto, se o autovetor dominante for múltiplo, o algoritmo gera iterações que alternam entre os dois subespaços sem chegar nunca a satisfazer um critério de parada.

Entretanto, GIBSON *et al.* (2000) assumiu que a matriz é não degenerada, o que implica em que nenhum dos autovalores é repetido. A situação real de repetição de um autovalor não é genérica. Entretanto, se eventualmente nosso algoritmo não convergir para

uma configuração estável, pode-se perturbar a matriz inicial, sem alterar a essência da informação lá contida, mas eliminando a degeneração da configuração inicial.

Assim, pode-se usar o STIRR no sistema aqui proposto sem maiores preocupações quanto a estas questões. O que este algoritmo efetivamente realiza quando se chega a uma configuração final é a separação dos valores em dois grupos de pesos facilmente separáveis por um algoritmo de classificação, posto que os elementos de um dos grupos geralmente têm um módulo significativamente maior que os elementos do outro. Assim, os dois grupos podem ser interpretados como os associados e os não associados ao elemento sobre o qual o sistema foi inicializado. Isto pode ser visto facilmente através de um pequeno exemplo artificialmente criado para demonstrar as capacidades do nosso algoritmo. Os dados consistem na tabela 4-4 a seguir.

Carro	**Nome**	**Cabelo**	**Categoria**
HONDA	Ricardo	Castanho	2
TOYOTA	Gilberto	Castanho	2
HONDA	Claudia	Preto	2
PEUGEOT	Sara	Louro	1
VOLKS	Paulo	Preto	1
PEUGEOT	Ines	Ruivo	1
TOYOTA	Victor	Preto	2
HONDA	Rafael	Branco	2
TOYOTA	Pedro	Preto	2
TOYOTA	Maria	Louro	2
FORD	Joana	Louro	1

Tabela 4-4: Dados gerados artificialmente para exemplo de aplicação do algoritmo STIRR

Os dados foram definidos de tal maneira que apenas aqueles elementos que têm um carro japonês (da marca HONDA ou da marca TOYOTA) pertencem à categoria 2. Todos os outros pertencem à categoria 1.

Ao executar o algoritmo usando uma inicialização aleatória e operador somatório para atualização, foram obtidos os seguintes pesos para cada um dos dados nas colunas do conjunto de dados, conforme mostrado na tabela 4-5.

Carro		**Nome**		**Cabelo**	
Valor	*Peso*	*Valor*	*Peso*	*Valor*	*Peso*
TOYOTA	0,515	Ricardo	0,091	Castanho	0,129
HONDA	0,291	Gilberto	0,091	Preto	0,516
PEUGEOT	0,099	Claudia	0,091	Louro	0,291
VOLKS	0,047	Sara	0,091	Ruivo	0,032
FORD	0,047	Paulo	0,091	Branco	0,032
		Ines	0,091		
		Victor	0,091		
		Rafael	0,091		
		Pedro	0,091		
		Maria	0,091		
		Joana	0,091		

Tabela 4-5: Resultados obtidos pela iteração do algoritmo STIRR sobre o exemplo descrito anteriormente.

Além dos valores obtidos para os dados, pode-se aplicar o mesmo algoritmo nas categorias a que eles pertencem, obtendo-se o valor de 0,092 para a categoria 1 (pessoas que possuem carros de qualquer nacionalidade que não a japonesa) e 0,907 para a categoria 2 (pessoas que possuem carros japoneses).

A separação entre estes dois grupos seria realizada facilmente por inspeção visual. Entretanto, para evitar intervenção humana neste passo do algoritmo, foi usado um simples algoritmo de K-Means para separar os dois conjuntos.

O K-Means é uma heurística de agrupamento não hierárquico que busca minimizar a distância dos elementos ao centro de clusters de forma iterativa. O algoritmo depende de um parâmetro (k=número de clusters) definido de forma *ad hoc* pelo usuário. Isto costuma ser um problema, tendo em vista que normalmente não se sabe quantos clusters existem *a priori*. Felizmente, neste caso específico sabe-se que existem exatamente dois clusters

definidos para cada variável, pois a separação em dois conjuntos é inerente ao algoritmo usado. Consequentemente, pode-se fixar o parâmetro k (número de conjuntos) em 2 sem nenhuma dúvida.

O algoritmo do K-Means pode ser descrito da seguinte maneira:

```
1. Escolher k distintos valores para centros dos grupos (possivelmente,
de forma aleatória)
2. Associar cada ponto ao centro mais próximo
3. Recalcular o centro de cada grupo
4. Repetir os passos 2-3 até nenhum elemento mudar de grupo.
```

Tendo em vista que os clusters usados aqui são bem definidos, com boa separabilidade, pode-se usar a norma euclidiana como métrica de distância de cada um dos pesos ao centro de seu cluster. A norma euclidiana é um caso especial de uma métrica de Minkowski (GORDON, 1981) e a sua fórmula para cálculo da distância entre dois objetos i, j é dada por:

$$d_{ij} = (\sum_{k=1}^{p} (x_{ij} - x_{jk})^{2})^{\frac{1}{2}}$$

Aplicando o K-Means nos valores dos pesos obtidos pelo algoritomo STIRR, os dados foram separados nos seguintes conjuntos:

- Carro: {TOYOTA, HONDA} e {VOLKS,PEUGEOT,FORD}
- Cabelo: {Preto, Louro} e {Branco, Ruivo, Castanho}
- Nome: Todos no mesmo conjunto, pois possuem valores iguais.

Estes conjuntos obtidos podem ser usados então como fonte de dados para usar em nossas regras, como será descrito mais adiante neste mesmo capítulo. Ao mesmo tempo, poderia se explorar o fato de que as categorias se separam em conjuntos de valores similares àqueles que as definem (a categoria 2 tem peso alto, como o conjunto {TOYOTA, HONDA}), mas isto é um caso especial derivado do fato de que existem apenas duas categorias e não generaliza bem para o caso de múltiplas categorias.

Quando se incluem dados numéricos nas tuplas, o problema complica-se um pouco, pois diferentes números podem ter exatamente o mesmo significado prático e não aceitar

uma conexão entre estes dados (figura 6-3) cria um sistema esparso demais incapaz de perceber as conexões que efetivamente existem entre os elementos

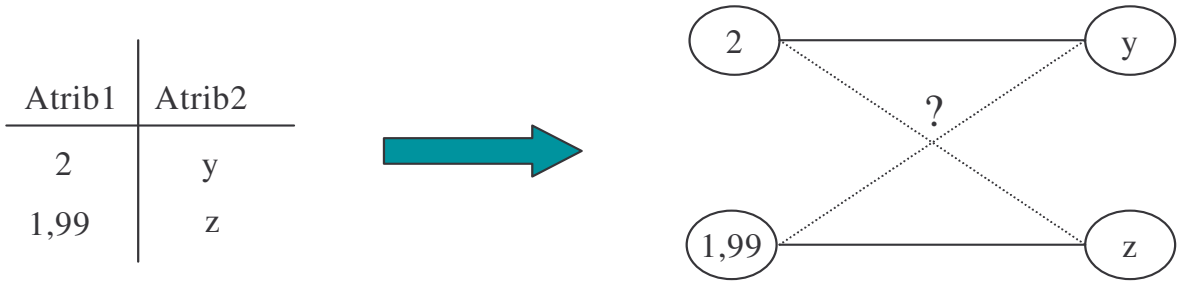


Figura 4-10 → Criação de grafo a partir de tuplas com dados categóricos e numéricos. Será que as linhas tracejadas não deveriam existir? Se a diferença de 0,01 entre os valores não for significativa, r ligações relevantes entre nós podem ser perdidas.

A solução para este problema é óbvia e consiste em discretizar as variáveis numéricas, de forma que valores próximos sejam transformados em um valor igual e o algoritmo possa proceder sem problemas.

Para discretizar uma variável é necessário definir o tamanho do intervalo entre os slots, que é dependente do valor absoluto das variáveis. Por exemplo, a variável salário, que vai de R$200 a R$10.000 não deveria ter um slot de intervalo 0,1, mas sim algo em torno de 50, enquanto que a variável altura, que vai de 1,5m a 2,1m deveria ter um slot de intervalo 0,05.

Uma solução alternativa consiste em fazer um arredondamento usando características dos conjuntos fuzzy, de tal forma que um valor seria fuzzyficado, as suas pertinências fuzzy seriam arredondadas e o valor seria defuzzyficado. Como o universo de discurso já varia de variável para variável, o tamanho do slot se ajusta automaticamente, sem necessidade de análises extensivas de variáveis.

Para começar o processo, divide-se o universo de discurso em *n* funções fuzzy, como vemos na figura 4-11. O uso de uma interseção por função é arbitrário – o algoritmo funciona igualmente bem com mais interseções.

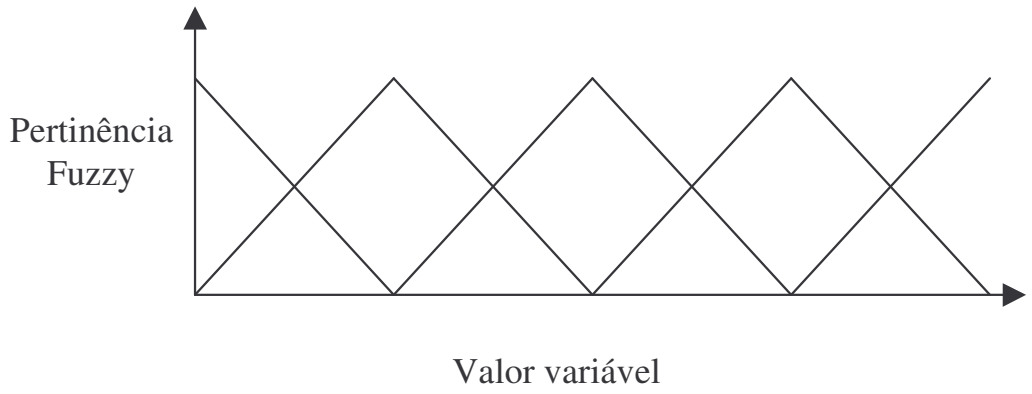


Figura 4-11 → Demonstração da divisão do universo de discurso de uma variável em conjuntos fuzzy para uso no método de arredondamento. O número de funções e interseções é determinado de forma ad hoc*, enquanto que os limites mínimo e máximo do universo de discurso da variável são lidos a partir dos dados.*

O número de conjuntos fuzzy utilizado não é relevante para o método, dado o seu objetivo final de integrar-se ao algoritmo de análise de dados categóricos. Os mesmos resultados são obtidos não importando o valor de *n* escolhido, visto que os valores serão discretizados sempre para o mesmo valor.

Para obter o valor desta discretização para um valor qualquer, este é fuzzyficado e são obtidas duas pertinências diferentes de zero (no caso de haver apenas uma interseção, como feito em todas as execuções do algoritmo). Estes valores são reais e devem então ser aproximados de forma arbitrária, como por exemplo para o centésimo.

Um exemplo possível: imagine que um determinado valor obtém pertinência 0,43381 em um conjunto em 0,56619 em outro. Aproximando para o centésimo, obteremos então uma pertinência de 0,43 no primeiro conjunto e 0,57 no segundo.

São defuzzyficadas estas pertinências aproximadas e obtêm-se o valores discretizado da variável em questão. Por exemplo, todos os números cuja pertinência no primeiro conjunto ficarem entre]0,425 e 0,435] terão sua pertinência aproximada para o mesmo valor e defuzzyficada da mesma forma, obtendo o mesmo valor discretizado.

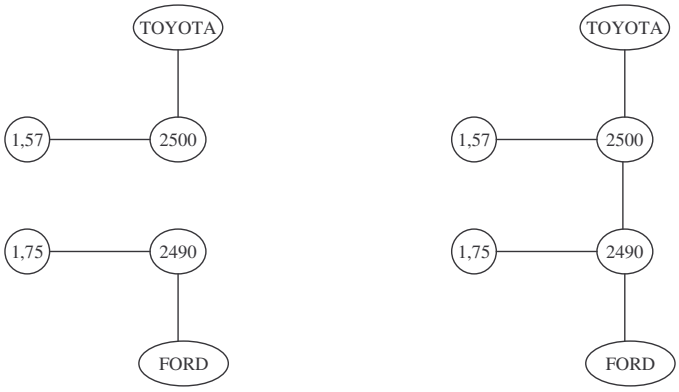


Figura 4-12 → Exemplo de geração do grafo definidos pelos dados quando utilizamos a técnica de arredondamento fuzzy.

Estendeu-se então o exemplo inicial, incluindo duas variáveis numéricas no conjunto de dados (salário e altura). Este conjunto de dados exemplo ficou então como mostrado na tabela 4-6 a seguir.

Altura	**Salario**	**Carro**	**Nome**	**Cabelo**	**Categoria**
1.75	2300	HONDA	Ricardo	Castanho	2
1.7	2000	TOYOTA	Gilberto	Castanho	1
1.7	3400	HONDA	Claudia	Preto	1
1.5	1800	PEUGEOT	Sara	Louro	2
1.8	2800	VOLKS	Paulo	Preto	1
1.65	1300	PEUGEOT	Ines	Ruivo	2
1.7	2100	TOYOTA	Victor	Preto	2
1.85	5000	HONDA	Rafael	Branco	2
1.62	5000	TOYOTA	Pedro	Preto	2
1.57	2500	TOYOTA	Maria	Louro	2
1.75	2490	FORD	Joana	Louro	1

Tabela 4-6: Novo conjunto exemplo, com a inclusão de atributos numéricos.

As categorias foram geradas usando-se como base a regra de que alguém pertence à categoria 2 se possui um carro japonês ou tem menos de 1,7m de altura.

Com estes dados, pode-se ver um exemplo prático da aplicação deste método na figura 6-5, na qual é mostrado o que acontece com um trecho do grafo quando aplica-se o método de discretização fuzzy ao conjunto exemplo anteriormente citado. Dois salários muito próximos (2500 e 2490, correspondentes às duas últimas linhas de nossa tabela) geram dois conjuntos disjuntos de nós (figura a), apesar de ambos transmitirem o mesmo poder de compra e por conseguinte, a mesma informação. Ao usar o método de discretização (precisão de centésimos na pertinência), ambos são arredondados para o mesmo valor, o que gera o gráfico mostrado na figura 6-5b.

Ao executar o algoritmo STIRR usando uma inicialização aleatória, foram obtidos os pesos para cada uma das colunas do conjunto exemplo, conforme descritos na tabela 4-7.

Altura		**Cabelo**		**Carro**		**Salário**	
Valor	*Peso*	*Valor*	*Peso*	*Valor*	*Peso*	*Valor*	*Peso*
1,49	0,052	Branco	0,032	FORD	0,047	1263	0,067
1,57	0,052	Castanho	0,128	HONDA	0,290	1820	0,067
1,62	0,052	Louro	0,290	PEUGEOT	0,099	2043	0,067
1,66	0,052	Preto	0,516	TOYOTA	0,516	2155	0,067
1,70	0,473	Ruivo	0,032	VOLKS	0,047	2266	0,067
1,74	0,211					2489	0,267
1,80	0,052					2824	0,067
1,85	0,052					3381	0,067
						5054	0,027

Tabela 4-7: Resultados obtidos pela iteração do algoritmo STIRR sobre o exemplo expandido incluindo dados numéricos, conforme descrito anteriormente.

Repare-se que os valores que aparecem nas colunas altura e salário não consistem nos valores originais, mas sim daqueles decorrentes do processo de arredondamento fuzzy aplicado.

Assim como no caso anterior, nome, que é uma variável unívoca, obtém pesos iguais para todos os elementos. Isto é coerente com o fato de que não se pode realizar

nenhum tipo de classificação baseada em chaves de bases de dados, dado que novos registros possuirão chaves unívocas nunca antes vistas em conjuntos de treinamento.

Aplicando o K-Means nos valores dos pesos obtidos pelo algoritmo STIRR, os dados foram separados nos seguintes conjuntos:

- Carro: {TOYOTA, HONDA} e {VOLKS,PEUGEOT,FORD}
- Cabelo: {Preto, Louro} e {Branco, Ruivo, Castanho}
- Salário: {1263, 1820, 2043, 2155, 2266, 2824, 3381} e {2489, 5054}
- Altura: {1.49, 1.57, 1.62, 1.66, 1.80, 1.85} e {1.70, 1.74}

O conjunto altura foi separado de forma espúria, colocando dois elementos maiores do que 1,7m dentro do conjunto dos pertencentes. Isto é decorrente do fato de que ambas as alturas aparecem em apenas um registro cada, no qual está definido um carro de marca japonesa. Para demonstrar este fato, foram acrescentadas ao arquivo de dados as linhas descritas na tabela 4-8.

Altura	**Salario**	**Carro**	**Nome**	**Cabelo**	**Categoria**
1.85	2000	VOLKS	Sicrano	Castanho	1
1.8	2000	FORD	Beltrano	Ruivo	1

Tabela 4-8: Linhas acrescentadas no arquivo para demonstrar efeito de uma má amostragem do conjunto de dados. Se houver várias linhas espúrias, estas podem alterar os conjuntos obtidos. É fundamental para a boa execução do algoritmo que os dados utilizados sejam representativos do conjunto como um todo, incluindo todas as classes e exemplos.

Omitem-se as pertinências para todas as outras variáveis que não altura por questão de concisão e de falta de necessidade, tendo em vista que nenhum dos conjuntos fuzzy obtidos para estas se alterou. As novas pertinências obtidas para a variável para a altura foram aquelas descritas na tabela 4-9 a seguir.

Altura	
Valor	*Peso*
1,49	0,040
1,57	0,040
1,62	0,040
1,66	0,040
1,70	0,359
1,74	0,160
1,80	0,160
1,85	0,160

Tabela 4-9: Novas pertinências obtidas para o conjunto altura.

Aplicando o K-Means nos valores dos pesos obtidos pelo algoritmo STIRR para o campo altura, os dados foram separados então nos seguintes conjuntos:

- Altura: {1.49, 1.57, 1.62, 1.66} e {1.70, 1.74, 1.80, 1.85}

Esta divisão é aquela corretamente preconizada pela estrutura de separação dos conjuntos propostos (que diz que todos os indivíduos que têm uma altura menor que 1,7m deve pertencer ao conjunto 2). Isto implica em que o conjunto de dados de treinamento deve ser representativo do conjunto de dados como um todo. Se houver algum tipo de *bias* na definição do conjunto de treinamento, isto se refletirá imediatamento nos conjuntos descobertos pelo algoritmo proposto.

Os elementos pertencentes aos conjuntos acima são os dados numéricos após o arredondamento. É importante que se aplique o arredondamento nos conjuntos de validação antes de aplicar as regras obtidas através deste algoritmo.

Como o universo de discurso dos conjuntos fuzzy é dependente dos valores assumidos pela variável, o método é auto-escalonável e não é necessário preocupar-se em determinar o tamanho dos slots de acordo com os valores assumidos por cada uma delas.

Isto faz com que se tenha menos um ponto de interferência por parte do usuário, diminuindo o número de parâmetros livres do sistema.

4.9 Algoritmos Genéticos Paralelos

4.9.1 Conceitos

A idéia por trás dos programas paralelos é dividir uma tarefa em sub-tarefas e resolver a todos eles simultaneamente usando múltiplos processadores. Esta estratégia de "dividir para conquistar" pode ser aplicada aos algoritmos evolucionários de várias maneiras diferentes e podemos encontrar várias referências a implementações paralelizadas de algoritmos evolucionários (DORIGO *et al.*, 1992, CANTÚ-PAZ, 1997)

Alguns métodos paralelos usam uma única população, enquanto outros dividem a população em várias sub-populações relativamente isoladas. Alguns métodos são mais adequados para arquiteturas massivamente paralelas, enquanto outros são mais adequados para computadores com poucos mas poderosos processadores.

A implementação paralelizada que descrevemos nesta seção é inspirada em conhecimento biológico bem difundido. Os biólogos descobriram que em ambientes isolados os animais são geralmente mais adaptados às peculiaridades do seu ambiente do que em grandes massas contínuas.

O primeiro a perceber isto foi o próprio Charles Darwin, que em sua viagem no navio H.M.S. Beagle percebeu ao passar por algumas ilhas, que as aves eram mais similares de uma ilha para outra do que os animais terrestres, já que estes não podiam mudar de ilha e por conseguinte apresentavam diferenças sutis que os tornavam mais adaptados às peculiaridades de cada ambiente.

Darwin percebeu que isto provavelmente decorria do fato de que as aves podem ir de uma ilha para outra facilmente para encontrar parceiros sexuais e difundir seu potencial cromossômico, enquanto que, por mais próximas que as ilhas estivessem, os animais terrestres estavam irremediavelmente presos nelas, sem a possibilidade de misturar seus genes com os de animais das outras ilhas.

Como conseqüência, cada população insular manteve suas características, e esta manutenção consiste na essência da teoria dos nichos ecológicos. Esta teoria, por sua vez inspirou a criação de novos modelos de algoritmos evolucionários paralelos.

DORIGO *et al.* (1992) teorizou que em um ambiente computacional pequenas populações separadas seriam mais competentes na busca de soluções através de um algoritmo evolucionário do que uma grande população unificada cujo tamanho fosse a soma dos tamanhos das pequenas populações.

Assim, desenvolveu-se um modelo de algoritmo evolucionário denominado *Island*, no qual a população de cromossomos é particionada em sub-populações isoladas cada uma das quais evoluirá de forma separada tentando atingir um determinado objetivo (maximização de uma função).

Periodicamente, cada população envia suas melhores *n* soluções para seus vizinhos, que descartam suas *n* piores soluções e as substituem pelos padrões recebidos. Esta troca é denominada *migração* e pode ser vista na figura 4-12.

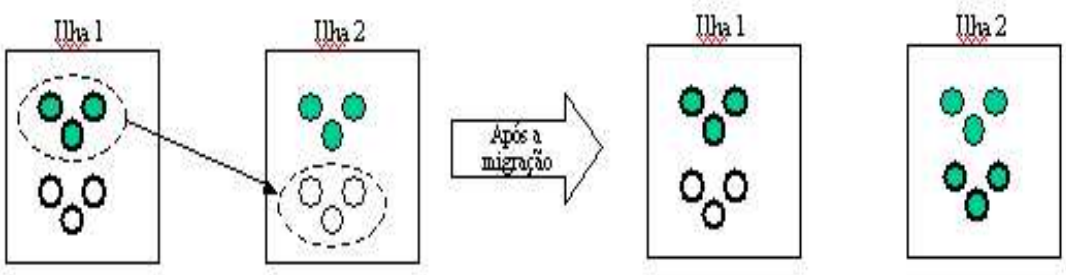


Figura 4-12 → Descrição do processo de migração de um algoritmo evolucionário paralelizado do tipo Island.Os melhores indivíduos da Ilha 1 (preenchidos) são selecionados para migração para a Ilha 2. Como a população é de tamanho fixo, os piores indivíduos da Ilha 2 (os sem preenchimento) são substituídos pelos elementos que chegam. O processo é repetido no sentido inverso (migração da Ilha 2 para a Ilha 1).

Este tipo de algoritmo evolucionário paralelizado é usado em ambientes distribuídos que tenham uma memória privada local. Dado que não existe um processo central de seleção, o sistema é completamente assíncrono, com os processadores se comunicando durante os intervalos de migração.

A utilização deste algoritmo permite que se criem várias populações com populações iniciais distintas que evoluem para tentar atingir o mesmo resultado. Após um determinado número de iterações, cada população sofrerá o efeito da convergência genética que será combatido utilizando-se o processo de migração.

Tendo em vista que cada população é inicializada de forma aleatória, esta tende a ser diferente da população de outro ambiente, e, por conseguinte, o processo migratório traz cromossomos que introduzem uma maior variabilidade para a população que o recebe.

Os elementos introduzidos pela migração possuem uma boa avaliação, já que eram os melhores indivíduos da população que os enviou. Isto quer dizer que eles provavelmente serão escolhidos durante o processo de aplicação de operadores para a criação de uma nova geração.

Aplicando-se esta paralelização pode-se executar várias rodadas de uma única vez. Entretanto, estas rodadas não executarão isoladamente, como seria o caso se as fizéssemos seqüencialmente em um único processador, mas sim cooperarão entre si para a obtenção de um resultado geral melhor.

4.9.2 Implementação

No caso do algoritmo desta tese, foi adotada a estratégia de dividir o processamento em vários computadores pessoais. Na maioria do tempo fora usados três deles, o melhor sendo um Pentium IV 1,7 Mhz com 256 Mb de memória.

Como os processadores usados eram diferentes, havia duas estratégias possíveis a adotar para o momento em que uma das ilhas atingissem o ponto de migração: esperar que as outras ilhas atingissem o mesmo ponto e sincronizar o processo ou continuar sem fazer a migração até alguma população estrangeira estar disponível.

Foi adotada a primeira opção, por questões de simplicidade e pelo fato de todos os computadores serem muito parecidos (o pior deles é um Pentium IV 1.2 Mhz), o que faz com que o tempo de espera seja relativamente pequeno.

A implementação da comunicação do aplicativo foi feita criando-se um servidor de memória central, que armazena os indivíduos migrantes até que outra ilha esteja pronto a recebê-los e a comunicação entre as ilhas e o pool de memória é feita usando-se o protocolo da Internet (TCP-IP via sockets).

Esta paralelização permite que o aplicativo seja dividido por um número grande de computadores, sem nenhuma restrição formal. No caso extremo, poderia se criar uma versão que fosse ativada em máquinas de pessoas interessadas em cooperar quando não estivessem usando seu micro. Estas versões poderiam rodar por um tempo pré-determinado ou até que os usuários retornassem ao seu computador. Assim, se poderia, ao menos teoricamente, infinitas ilhas de processamento que poderiam colaborar para resolver qualquer problema, por mais difícil que este se mostrasse.

Todas as execuções descritas no próximo capítulo foram feitas utilizando-se mais de um computador simultaneamente. Em cada micro foi evoluída uma população independente, cujos 3 melhores indivíduos migravam a cada 10 gerações para as outras populações em evolução.

Cada micro usado possui um processador e uma quantidade de memória distinta, e todas as populações evoluem de forma assíncrona. Isto implica em que um processador chega ao momento de migração antes dos outros. Para que este micro com processador mais veloz não tenha que parar para esperar pelas outras populações, foi criado um banco central de indivíduos. Ao fim de cada execução o programa checa com este banco central se existem migrações a efetuar, e havendo, ele as recebe, substituindo seus 3 piores cromossomos pelas soluções recebidas.

4.10 Comparando a representação proposta com outras pré-existentes

Nesta seção será comparado o uso prévio da notação polonesa reversa (RPN) em vários artigos que também usam algoritmos evolucionários com a representação dos cromossomos similares àquela usada nesta tese.

YANG *et al.* (2003) busca desenvolver regras para análise de pontos de clivagem de proteases. Esta representação não utiliza conjuntos fuzzy, pois é voltada para regras que analizam a presença de aminoácidos (um conjunto de 20 letras). Assim, o único formato necessário de regra consiste em combinação de avaliações do tipo <*posição*> <*operador*> <*aminoácido*>, onde <*operador*> pode ser igual ou diferente.

DASGUPTA *et al.* (2002) usa uma representação que se baseia em string de bits para resolver o problema de detecção de intrusos em uma rede. Assim como no caso da representação adotada no capítulo anterior para o caso binário, esta solução tem o problema de não ser escalável e ter cromossomos de tamanho praticamente intratável quando do crescimento do número de parâmetros. No caso do trabalho mencionado isto não é um empecilho visto que o número de atributos sendo tratados é pequeno (da ordem de poucas dezenas).

O algoritmo evolucionário descrito aqui difere destes dois trabalhos citados no fato de que o operador usado nesta tese, que é fortemente baseado nos operadores padrão de programação genética (KOZA, 1992), é similar ao crossover uniforme, enquanto que ambos os trabalhos usam uma breve variação do crossover de um ponto, que é reconhecidamente inferior na preservação dos esquemas e das características interessantes dos cromossomos (MITCHELL, 1996).

BENTLEY (1999), de forma semelhante ao trabalho desenvolvido nesta tese, busca desenvolver bases de regras fuzzy para análise de padrões. Este artigo, tal como (DASGUPTA *et al.*, 2002), usa uma representação baseada em strings de bits que, como

discutido anteriormente, faz com que o cromossomo se torne intratavelmente grande em problemas com um número elevado de atributos.

Outro ponto que merece destaque nestes artigos é o fato de que eles geram seus conjuntos fuzzy sem interseção. Para poder realizar uma comparação justa, foi tentado usar este tipo de definição em nosso algoritmo e descobriu-se que a desempenho do previsor/classificador era inferior àquela obtida usando-se conjuntos com interseção, dado o grande número de regras que não disparavam e a falta de precisão resultante do módulo fuzzy.

Obviamente não é possível dizer que isto ocorre também nos trabalhos citados tendo em vista o fato de que nesta tese usam-se conjuntos de dados radicalmente diferentes. Entretanto, é de se esperar que este tipo de melhoria quando do uso de conjuntos fuzzy com interseção seja verificado em todas as aplicações.

ZHOU *et al.* (2002) usa algoritmos evolucionários para criar uma base de regras descritas com RPN. Entretanto, neste artigo usam-se operadores matemáticos tais como raiz quadrada, adição e multiplicação que apesar de serem interessantes para a aplicação deste artigo, não fazem sentido em um contexto de regras fuzzy.

Alguns destes operadores também não têm sentido nas aplicações biológicas que serão descritas nos próximos capítulos. As células não realizam operações matemáticas com a concentração de substâncias. As operações celulares são processos estocásticos cuja probabilidade é diretamente proporcional à presença (ou ausência) dos elementos promotores (ou inibidores) (ALBERTS *et al.*, 2002).

Se dois promotores são necessários, pode-se interpretar como se estivesse sendo efetuado algum tipo de multiplicação ou convolução. Entretanto, posto que não existem pequenos somadores/integradores dentro da célula, seria uma modelagem mais precisa considerar esta necessidade como sendo uma pré-condição, como é no contexto fuzzy e definido no algoritmo proposto nesta tese.

Capítulo 5 - Aplicações em bioinformática

Existem várias aplicações possíveis na área de bioinformática e a escolhida para a aplicação do algoritmo desta tese foi a análise de microarrays de DNA, que possui algumas características especiais. A principal delas consiste no fato de que estes microarrays possuem informação sobre um grande número de genes medidos em um pequeno número de instantes de tempo. Isto decorre do fato das medições serem feitas por microarrays e serem financeiramente onerosas e faz com que as matrizes de dados tenham um número de colunas da ordem dos milhares e um número de linhas da ordem das unidades.

Outra característica importante consiste na grande quantidade de informações parciais disponíveis hoje em dia na área de biologia. A utilização destas informações é imprescindível para a obtenção de modelos precisos que tenham um bom poder de previsão.

Este capítulo será iniciado com uma descrição do conhecimento biológico mínimo necessário para a compreensão do problema. Na seção a) serão discutidas as redes de regulação genética e na seção b) os microarrays de DNA e as principais características dos dados obtidos através deles. Uma vez estabelecidos estes conhecimentos, será descrita a aplicação do algoritmo proposto nesta tese com detalhes, especialmente quanto às necessidades de pré-processamento e a incorporação de conhecimento pré-existente.

5.1 Pré-processamento dos dados

Este passo é necessário para diminuir a dimensionalidade do nosso conjunto de dados quando trabalhamos com conjuntos provenientes de microarrays. Nestes casos, temos conjuntos de dados de dimensões n X m, onde n é da ordem de 10^3 enquanto que m é da ordem de 1. Obviamente tal conjunto não pode ser tratado usando-se técnicas estatísticas tradicionais tais como análise de convariância (MOORE *et al.*, 2002).

Se for possível diminuir a dimensionalidade dos dados, torna-se mais fácil para qualquer algoritmo encontrar soluções com base nos dados. Assim, é interessante que se use técnicas estabelecidas, baseadas em estatística ou conhecimento biológico prévio, para diminuir ao máximo as dimensões dos dados antes de oferecê-los ao algoritmo aqui proposto. Quando outros problemas que não possuirem esta grande discrepância entre as dimensões forem tratados, este passo pode vir a ser dispensável.

Nos problemas biológicos, primeiramente eliminam-se os elementos que não podem ter função de controle. Isto decorre do fato de se estar procurando por elementos que possam ser candidatos a reguladores. Esta aplicação de conhecimento biológico prévio permite que o espaço de busca original seja reduzido aos fatores de transcrição, quinases de proteína, transdutores de sinal e proteínas de ligação.

É fácil entender este passo quando se pensa que se está buscando elementos que podem ter um papel na regulação e que apresentem mudanças significativas nos seus níveis de expressão, de forma a serem detectados através de microarrays. Alguns elementos, como fatores de transcrição, agem mesmo quando presentes em níveis muito baixos (às vezes, basta uma molécula por célula). Entretanto, outras moléculas, que são diretamente influenciadas por estes fatores podem estar presentes em níveis muito mais significativos, o que significa que mesmo que não se encontre o controle real, encontraremos uma ligação de controle epistático (PE'ER *et al.*, 2002).

Os conjuntos de dados usados nesta tese são aqueles gerados por microarrays e que representam o logaritmo da variação da expressão. Entretanto, dado que a variação não está linearmente relacionada com a intensidade do sinal é possível superestimar esta variação nas baixas intensidades ou subestimá-la quando o elemento em estudo estiver presente em grandes quantidades.

Para representar isto imagine que existam dois elementos: o primeiro elemento varia seu nível de expressão de 0,0001 para 0,001 (variação de ordem de 10) enquanto que o segundo elemento varia de um nível de expressão de 10 para um nível de 20 (uma mudança de ordem 2). Apesar da segunda mudança ser muito mais importante e estatisticamente significativa que a segunda (que pode se dever apenas a ruído ou erros de medição), ela será considerada menos importante, tendo em vista que a magnitude da variação é menor do que no primeiro caso.

Por conseguinte, baseando-se apenas nas variações de intensidade, serão obtidos um número possivelmente grande de falsos positivos e falsos negativos para genes cujos níveis de expressão sejam muito baixos ou muito altos, respectivamente. Esta é uma limitação do algoritmo da qual é necessário estar consciente.

Para minimizar este efeito, são removidos do banco de dados aqueles elementos que têm um nível de expressão muito baixo ou cujas variações de nível de expressão são não significativas, podendo ser atribuídas a fatores tais como erro aleatório de medição, conforme processo manual descrito abaixo.

Esta remoção não está relacionada com o fato dos elementos serem considerados como “outliers”, mas sim com os conceitos biológicos associados à obtenção dos dados de microarray.

Primeiramente, quando a expressão de um gene é muito baixa, existem problemas de medição pois está é feita em termos de luminosidade. Ademais, tendo em vista que os spots (local de cada gene) não são perfeitamente limitados geometricamente, é possível que a superexpressão de um gene vizinho afete ligeiramente a expressão de um gene. Assim, se esta expressão já era inicialmente baixa, este efeito pode afetar significativamente o valor obtido.

Segundo, quando a expressão de um gene é muito baixa, certos efeitos estocásticos do comportamento de moléculas de DNA passa a ser significativo. Existe uma tendência a

pensar que duas seqüências complementares de DNA sempre se ligam, mas isto não é verdade. Esta ligação ocorrerá com uma probabilidade determinada que é afetada por vários valores, entre os quais a temperatura do ambiente. Logo, se houver um baixo nível de expressão, este pode ser afetado por estes fatores estocásticos e ter uma significância menor.

Estes dois fatores justificam o trabalho de eliminação dos elementos de baixo nível de expressão. Este trabalho é feito de forma manual, usando-se conhecimento prévio dos biólogos, pois cada gene tem um nível distinto de expressão significativa. Para cada gene deve ser estabelecido se seu nível de expressão é significativo, com base em informações sobre seus níveis normais e nas variações esperadas quando ele está expresso.

Além disto, levando-se em consideração que os microarrays utilizados não medem o nível absoluto de expressão, mas sim a variação da expressão relativa a um valor basal, podemos ter variações não significativas, também devido aos valroes discutidos acima.

Assim, em discussões com biólogos, estabeleceu-se de forma *ad-hoc* um limite mínimo de 0,3 para as variações de expressão a serem processadas pelo algoritmo. Este limite corresponde a uma duplicação dos níveis de expressão, mudança esta que necessariamente não pode se considerar como conseqüência de efeitos estocásticos, posto que quando se aplica este limite, já se eliminou todos os elementos com níveis de expressão pouco significativos.

Para efeitos ilustrativos, mencionamos que ao terminar este passo, no caso do maior problema estudado nesta tese, o da resposta ao frio da *Arabidopsis thaliana*, a base original que tem tamanho de 8000 X 7, passa a ter dimensões de 176 genes medidos durante 7 instantes de tempo, que não são tratáveis por métodos estatísticos tradicionais, mas que já são perfeitamente adequadas para tratamento pelo algoritmo proposto nesta tese.

Este trabalho foi realizado na base completa, pois como o instante t é usado para prever o instante $t+1$, se o instante t for retirado, não haverá como prever os valores de expressão do estado $t+1$, inviabilizando testes do tipo *n-fold*.

5.2 Separando candidatos a regulador

Tendo em vista a alta dimensionalidade dos dados, seria interessante reduzir o escopo de nossa busca, restringindo-a apenas a aqueles genes que se mostrem os candidatos mais fortes a serem reguladores do gene em questão. Para tanto pode ser usada a correlação entre as expressões.

A correlação não necessariamente implica em causalidade. Para determinar que existe causalidade a partir de uma alta correlação, é necessário estabelecer que as mudanças na variável que é dita de causa antecedem no tempo as mudanças da variável efeito, além de adequar a hipótese da causalidade com o modelo teórico específico do fenômeno sendo estudado.

Para determinar que a variável de causa antecede a variável de efeito, usa-se a medida da correlação com atraso (MOORE *et al.*, 2002). Esta se refere à correlação entre o nível de expressão do gene de interesse em um determinado instante *t* (dito gene regulado) com os níveis de expressão de todos os outros genes nos instantes t-Δt, $\forall\Delta t=0,1,2,....$ Isto permite reduzir o número de candidatos a regulador, excluindo todos aqueles que não demonstram possuir um relacionamento significativo com o gene regulado.

É importante entender que altas correlações também podem ser causadas por uma resposta comum de duas variáveis a uma terceira que não esteja incluída no modelo. Ademais, uma correlação baixa não necessariamente implica na ausência de causalidade, devido ao efeito da confusão (MOORE *et al.*, 2002). Este efeito é causado por uma terceira variável, escondida ou não, que afeta a ação do gene regulador sobre o gene regulado, modificando o resultado da regulação. Estes efeitos estão resumidos na figura 5-5.

Estes fatores levam à conclusão de que, ao usar esta técnica, corre-se o risco de perder relacionamentos significativos entre genes. Entretanto, é um risco aceitável dado o benefício que se terá ao obter um algoritmo que rode em um tempo computacionalmente viável, além da própria questão da significância dos resultados. Tendo em vista que o número de pontos (dimensão coluna da matriz de dados) é limitado, é importante limitar-se o número de genes analisado (dimensão linha da matriz de dados).

Se a ordem de grandeza da matriz for drasticamente reduzida, pode-se até considerar a utilização de métodos estatísticos. Mesmo que estes métodos ainda não possam ser utilizados devido à falta de dados, a limitação do número de elementos diminui o número de graus de liberdade fazendo com que os resultados obtidos sejam mais significativos.

Uma das maneiras com que se busca diminuir o número de falsos positivos que esta hipótese fundamental pode gerar consiste em agrupar os elementos por nível de expressão de forma a buscar relacionamentos significativos para vários elementos que supostamente estejam sob o mesmo tipo de controle, conforme será descrito na seção a seguir.

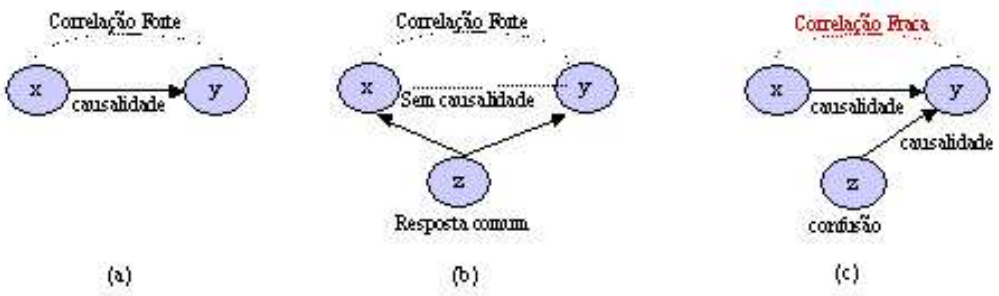


Figura 5-5: Possíveis relacionamentos entre variáveis, de acordo com o nível de correlação apresentado.

5.3 Agrupamento

Agrupamento (clustering) consiste em particionar um determinado conjunto de dados em grupos baseados em uma métrica baseada em uma ou mais características dos dados de tal forma que todos os pontos em um determinado conjunto sejam mais similares aos outros elementos do mesmo grupo do que a qualquer outro elemento de outro grupo do universo.

Várias abordagens já foram propostas para este problema, usando-se diferentes métricas e diferentes técnicas e algumas destas foram aplicadas a dados de expressão gênica, se mostrando úteis para identificar agrupamentos biologicamente significativos de

genes e amostras, assim como para responder algumas questões sobre função de algum gene, regulação e expressão diferenciada de genes em várias condições distintas.

Algoritmos de agrupamento se mostraram úteis para agrupar genes de funções similares baseando-se nos padrões de expressão em várias amostras distintas, sejam de tecidos ou de condições diferentes. Genes co-expressos encontrados no mesmo cluster normalmente compartilham função ou se apresentam no mesmo caminho de regulação (*pathway*).

Usando-se técnicas de agrupamento é possível juntar genes pouco conhecidos com famílias de outros já bem estudados, o que pode ajudar a entender e/ou teorizar sobre as funções de vários genes para existe pouca informação disponível a priori (JIANG *et al.*, 2002).

A abordagem discutida a seguir é baseada na hipótese de que genes que exibem o mesmo comportamento estão sob o mesmo tipo de controle. Quando se lida com um número limitado de medidas, pode-se acreditar nesta hipótese de forma errônea.

É possível que dois genes exibam um mesmo comportamento em um intervalo de tempo limitado, dado um número pequeno de medidas (em instantes de tempo distintos), mesmo que submetidos a estratégias de regulação diferentes. Isto decorre do fato de que duas diferentes estratégias de regulação podem partir de um mesmo ponto e gerar a mesma resposta.

Um exemplo trivial de tal problema consiste em observar dois genes que tenham as seguintes estratégias de regulação:

$$gene_1 = \frac{x(t-1) + y(t-1)}{2}, x(0) = 1$$

$$gene_2 = 1, \forall t \geq 0$$

Se for feito um acompanhamento da evolução dos dois elementos, os genes 1 e 2, poderá se perceber que ambos se mantém constantes em 1. Um algoritmo de agrupamento poderia supor que ambos estão sujeitos à mesma estratégia de regulação, o que obviamente não é verdade. Este tipo de inferência errada poderia ser desfeita facilmente se fossem obtidos mais trajetórias, com valores iniciais diferentes para o $gene_1$, o que pode não ser

possível em situações reais. Isto quer dizer que esta hipótese deve conduzir a uma estratégia que fornecerá alguns falsos positivos (aqueles para os quais parece haver uma estratégia de regulação compartilhada, apesar de não haver).

Por outro lado, não há nenhuma razão para supor que tal estratégia nos conduza a obter falsos negativos. Isto quer dizer que infelizmente as hipóteses geradas pelo algoritmo aqui proposto não serão necessariamente todas consistentes, podendo haver a inferência equivocada de participação de certos elementos em alguns caminhos regulatórios.

É importante lembrar, entretanto, que o objetivo proposto nesta aplicação é a obtenção de um número pequeno de hipóteses plausíveis. A presença de falsos positivos significa que será necessário um trabalho adicional para eliminá-los usando-se informações e/ou experimentos adicionais.

Uma métrica freqüentemente utilizada para indicar a similaridade entre os níveis de expressão e suas mudanças é o coeficiente de correlação (r) que é uma medida do grau de relacionamento linear entre duas variáveis (geralmente chamadas de X e Y).

Enquanto que na regressão a ênfase era na predição de uma variável baseando-se em outra, na correlação a ênfase é em detectar até que ponto um modelo linear pode descrever o relacionamento entre duas variáveis. Na regressão, o interesse é direcional (há uma variável predita e outra previsora), enquanto que na correlação o interesse é não direcional, sendo o relacionamento o aspecto crítico a modelar (STOCKBURGER, 1996)

Usar valores de correlação também é interessante, pois pode gerar grupos com interseção. O fato de que o gene1 está correlacionado com o gene2 em um certo grau e este está correlacionado com o mesmo grau a um outro gene (gene3), não implica que este último e o primeiro (gene1) estejam relacionados com este mesmo grau.

Um exemplo deste tipo de falta de transitividade da correlação pode ser vista através de um conjunto de dados de valores de expressão criados artificialmente e sem relação com exemplos anteriores, mostrados na tabela 5-1, cujo gráfico pode ser visto na figura 5-6.

gene1	gene2	Gene3
1	1,6	1,2
2	2,3	2,1
3	3,4	3,1
4	3	2,5
6	4,9	4,7

Tabela 5-1: Exemplo de valores de expressão criados artificialmente

As correlações entre estes elementos são as seguintes:

r(1,2)=0,955

r(2,3)=0,995

r(1,3)= 0,934

Se for estabelecido como limite mínimo de correlação entre os elementos o valor de 0,95, serão criados dois grupos para estes três genes, grupos estes cujos componentes são respectivamente {1,2} e {2,3}.

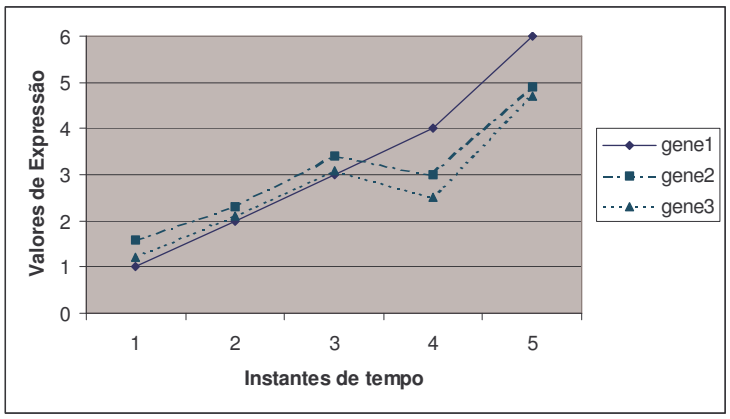


Figura 5-6: O gráfico dos genes fictícios do exemplo. Podemos ver que as séries dos genes 2 e 3 mudam de forma quase idêntica, o que justifica o valor de correlação muito próximo de 1.

Logo, pode-se criar um cluster para cada elemento da matriz, contendo todos os elementos que estão fortemente correlacionados a um elemento de interesse, a ser determinado pelo pesquisador que está usando este algoritmo.

Como pode ser visto no simples exemplo dado acima, estes clusters não serão necessariamente disjuntos, o que é perfeitamente condizente com o fato já mencionado

anteriormente de cada gene estar envolvido em até dez diferentes processos regulatórios dentro de uma célula (ARNONE *et al.*, 1997, ALBERTS *et al*., 2002).

Neste momento o objetivo principal é diminuir a dimensionalidade do problema e descobrir estratégias de controle. Baseado na nossa premissa de que genes que apresentam o mesmo comportamento estão sob o controle do mesmo processo regulatório, pode-se descobrir, usando técnicas de agrupamento, todos os genes que possuem o mesmo comportamento do gene de interesse.

Usando-se a premissa fundamental, pode-se afirmar que qualquer estratégia descoberta que funcione no gene de interesse também deve funcionar em todos os genes pertencentes ao seu agrupamento.

Procurar então por estratégias de regulação que funcionem em todo o grupo de genes ajuda a eliminar estratégias que funcionem bem apenas nos dados do gene de interesse. Esta eliminação, por sua vez, fará com que os resultados obtidos sejam mais próximos de estratégias que efetivamente se mostrem boas candidatas para teste em laboratório.

A técnica usada para separar os elementos que apresentam o mesmo padrão de expressão consiste em determinar qual é o elemento de interesse (aquele que se pretende estudar) e depois estabelecer *ad hoc* um limite mínimo de correlação para que o elemento entre no grupo em consideração.

Este é um limite *crisp*, ou seja, elementos que estejam imediatamente abaixo deste valor de correlação não farão parte do grupo, apesar de seu valor de correlação poder ser muito próximo a outros que eventualmente estejam presentes no grupo.

Um exemplo deste caso seria se fosse estabelecida uma correlação de corte de 0,95. Elementos de correlação com o elemento em estudo igual a 0,9501 estariam presentes, enquanto elementos com correlação igual 0,9499 estariam ausentes. Este tipo de problema é inerente à técnica usada e não pode ser evitado.

É importante ressaltar que não se está falando do valor absoluto da correlação, mas sim de seu valor real. Os elementos com valores de correlação negativos e próximos de menos um estão quase que perfeitamente anti-correlacionados com o elemento de interesse e poder-se-ia imaginar que sua estratégia de controle fosse exatamente a inversão da

estratégia usada para o elemento de interesse (se este fosse ativado por uma substância, os elementos anti-correlacionados estariam inibidos pela mesma).

Entretanto, esta seria mais uma suposição *ad-hoc* e, no intuito de minimizar o número de hipóteses em que os resultados se baseiam, decidiu-se por não se incluir os elementos anti-correlacionados nos agrupamento obtidos.

Um exemplo de cluster gerado por nossa técnica está mostrado na figura 5-7. É importante perceber que apesar dos níveis de expressão dos diversos genes serem diferentes, os padrões de expressão de todos (a trajetória no tempo) é bastante similar.

Assim, para cada gene de interesse (o gene regulado) foi descoberto dentro do conjunto de dados de microarray um grupo de elementos cujas trajetórias são similares àquela do genes do interesse. Partindo-se do pressuposto de que genes com padrão de expressão similar estão sob a mesma regulação, pode-se então aplicar o algoritmo inteligente descrito nesta tese a todo o grupo simultaneamente.

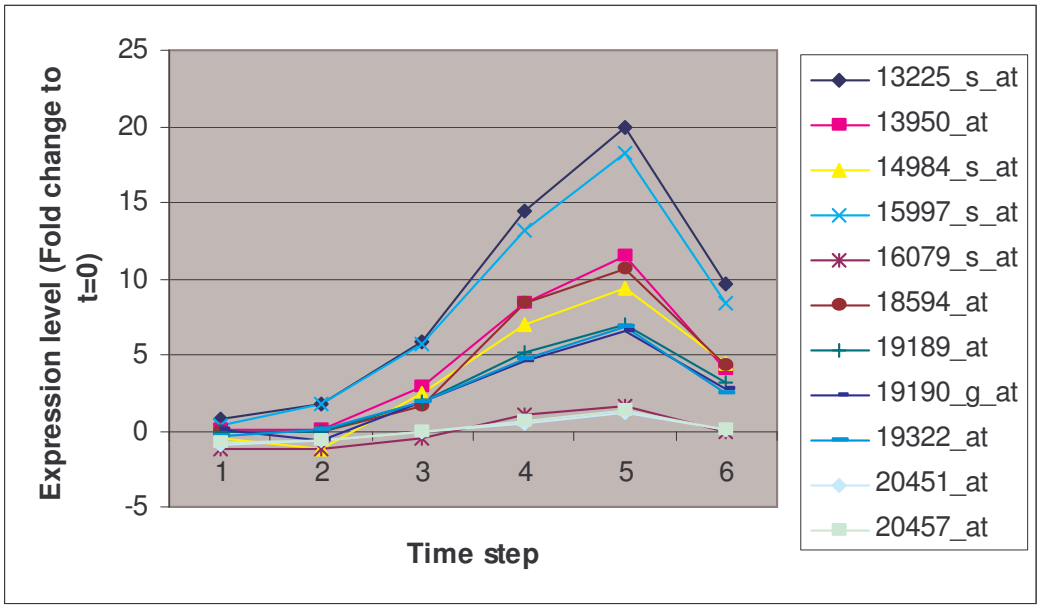


Figura 5-7: Um dos clusters gerados pela aplicação da técnica de agrupamento nos dados de expressão da Arabidopsis thaliana.*A abcissa corresponde ao instante da medida, enquanto que a ordenada corresponde ao nível de expressão dos genes. Cada linha corresponde ao padrão de expressão de um gene distinto, cujo nome está descrito na tabela ao lado do gráfico. Pode se perceber que, apesar dos níveis de expressão dos vários genes serem distintos, o formato da curva dos genes pertencentes a este grupo são semelhatnes, correspondendo a uma subida até o instante 5, quando o nível de expressão começa a cair.*

5.4 Incorporando conhecimento existente

Um grande problema da maioria das técnicas modernas de extração de dados a partir de experimentos biológicos (de microarray ou de outros) consiste no fato em a maioria destas técnicas que na maioria das vezes não incorporam nenhum ou quase nenhum conhecimento pré-existente, baseando-se apenas nos dados disponíveis (SCHRAGER *et al.*, 2002).

Um exemplo deste tipo de comportamento consiste nas técnicas de agrupamento, que simplesmente separam os dados de acordo com seus perfis de expressão, sem utilizar qualquer informação adicional.

A utilização exclusiva dos dados como fonte de descobertas não condiz com a maneira como os biólogos trabalham. Estes profissionais costumam apoiar-se firmemente no conhecimento pré-existente para desenvolver modelos e testar estes modelos contra os dados disponibilizados pelas técnicas modernas.

Este conhecimento pré-existente não pode ser desprezado. Em várias situações, existem modelos parciais de regulação já comprovados em diversos experimentos em laboratório. Assim, seria interessante usar os dados disponíveis em laboratório para testar e estender estas hipóteses criadas pelos biólogos.

Ademais, a busca cega pode ser interessante para se testar uma técnica para tentar determinar se ela consegue descobrir certos relacionamentos fundamentais previamente conhecidos, mas não é uma estratégia eficiente de descoberta de novos relacionamentos de uma rede de regulação genética, pois a existência de relacionamentos prévios cria pré-condições e limita o espaço de busca, permitindo que nós façamos uma busca muito mais eficiente do que se nós os ignorássemos.

Ademais, algoritmos de busca cega normalmente não incluem mecanismos para inclusão de conhecimento prévio para direcionamento da busca, o que faz com que normalmente este tipo de conhecimento seja simplesmente desprezado.

Assim sendo, nossa técnica permite a incorporação de conhecimento pré-existente, através da utilização de uma interface própria. Conforme veremos na seção de resultados, a incorporação de conhecimento prévio serve também como forma de validação de modelos biológicos existentes.

5.5 Resultados

Para testar nosso algoritmo, nada melhor do que usar conjuntos de dados reais e verificar os resultados contra conhecimento biológico pré-existente. Para fazê-lo, então, usamos dois conjuntos de dados: um referente à resposta ao frio por parte da planta *Arabidopsis thaliana* e outro que consiste na expressão dos genes de uma determinada família no desenvolvimento do sistema nervoso central de ratos.

Em todos os resultados mostrados a seguir foi usado um critério de terminação baseado no número de gerações (máximo de 80 gerações por rodada), estagnação (paramos o algoritmo se a melhor solução estagnasse por 10 gerações) e qualidade da solução (interrompeu-se a execução quando o erro do casamento dos dados fosse menor ou igual que 0.001).

É usada uma roleta para selecionar os operadores. O crossover recebeu uma probabilidade inicial de 0,95 enquanto que os três operadores de mutação receberam uma fitness igual a 1 – fitness do crossover.

Os operadores recebem probabilidades distintas para que se possa fazer com que o operador mais útil em determinado momento seja selecionado de forma mais frequente. No início da execução, enquanto a população é muito variada, é importante que o operador de crossover predomine, para combinar o material genético das melhores soluções. Entretanto, após um grande número de rodadas, a população sofrerá os efeitos da convergência genética, comum a populações fechadas que reproduzem, o que faz com que o operador de mutação seja mais proveitoso na melhoria genética da população.

Uma vez selecionado o operador de mutação, haveria um segundo sorteio para determinar qual dos três (alteração aleatória, inserção de regra ou exclusão de regra) seria

escolhido, na qual o operador de mutação de alteração aleatória de uma regra receberia uma chance de 60% enquanto que os outros dois operadores receberiam uma chance de 20% cada.

Uma vez escolhido, o operador de mutação de alteração aleatória alteraria cada ramo de uma árvore com probabilidade de 5%. Esta probabilidade é bastante alta para os padrões da literatura, mas esta normalmente utiliza o operador de mutação em tandem com o operador de crossover. No nosso caso eles estão competindo entre si e o operador de alteração aleatória ainda compete com outros operadores de mutação, o que diminui sobremaneira a taxa efetiva de mutação.

A probabilidade do operador de crossover foi colocada como linearmente decrescente até que na rodada final sua fitness fosse de 0,2 (a dos operadores de mutação se manteve linearmente crescente, sempre sendo equivalente a 1 – probabilidade$_{crossover}$, como é possível ver na figura 5-8).

Cada população consistiu de 100 indivíduos e foi usado um módulo de população elitista que transfere as melhores 2 soluções da geração atual para a próxima e permite que as outras 98 soluções estejam sujeitas à substituição por indivíduos gerados através da aplicação dos operadores genéticos.

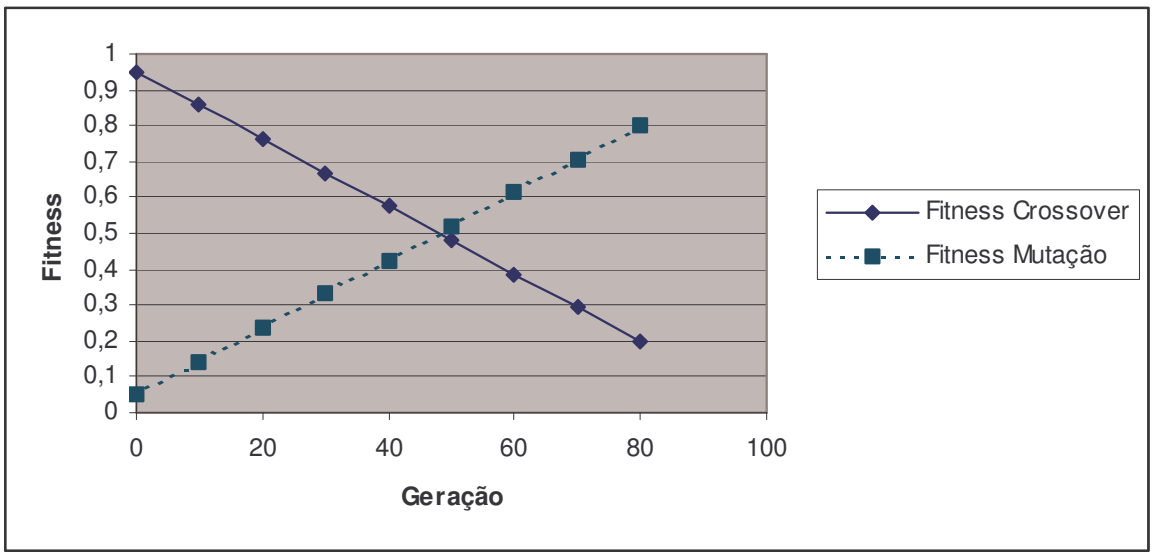


Fig 5-8 → Evolução da fitness dos operadores usados no nosso programa genético

Em ambos os exemplos descritos a seguir o algoritmo proposto foi executado 10 vezes e os dois melhores resultados de cada rodada foram usados como sementes de inicialização para a 11ª rodada, cujos resultados são os mostrados neste documento.

Esta abordagem pode ser considerada como "otimista", sendo que alguns poderiam apontar para o uso da média dos melhores resultados como uma métrica mais adequada. Entretanto, é praxe, no ramo dos AE, utilizar o melhor resultado entre aqueles obtidos, pois não faz sentido obter uma boa solução para o problema e utilizar outra solução pior.

Outro ponto a ressaltar é que quando se executam múltiplas rodadas de um AE, os valores de variância e desvio padrão do erro para um AE têm a tendência de serem elevados, dada a influência de fatores estocásticos na inicialização e evolução da população. Estes valores, por conseguinte, não agregam informação, e mesmo que sejam baixos, não são bons indicativos do desempenho do AE em um problema novo que proventura surja.

São mostrados os resultados da divisão do espaço de expressão (universo de discurso) de cada gene em 3 conjuntos fuzzy distintos. Foi tentada também a divisão do universo de discurso em 2, 5, 7, 11 e 15 conjuntos fuzzy. Entretanto, nenhuma melhora significativa nos resultados foi obtida. Nos casos de 5 regras ou mais, é provável que haja um over-fitting dos dados sem aprendizado do processo subjacente, sem contar que o número de regras com ativação zero cresce. Para cada conjunto foi permitido um máximo de 2 regras por conjunto fuzzy.

Posteriormente foi desenvolvido um mecanismo simplificador de pré-condições baseado no mecanismo de expressões regulares do PERL que corta expressões desnecessariamente longas (tais como A AND A) e as substitui pela sua forma mais simples (no caso do exemplo, A). As regras mostradas aqui já passaram por este mecanismo de simplificação, mas o algoritmo originalmente gera várias regras com características repetidas.

5.5.1 Resposta ao frio da *Arabidopsis thaliana*

O primeiro conjunto de dados que utilizamos foi gerado por experimentos de microarrays, consistindo na expressão de todos os genes da planta *Arabidopsis thaliana* (cerca de 8000 no total) em um número limitado de instantes de tempo (no nosso caso, 7 apenas). Nosso objetivo primoridal consistia em utilizar nosso algoritmo para realizar uma busca pela regulação de alguns elementos específicos dentro do conjunto de genes disponíveis, mais especificamente aqueles envolvidos na resposta desta planta ao frio.

É importante ressaltar um fato que já mencionamos anteriormente: que estes valores de dimensões são comuns no campo da biologia. Como os experimentos de microarrays analisam até milhares de dados simultaneamente, temos milhares de linhas em nossa matriz de dados. Entretanto, sua realização é cara, é comum que sejam realizados poucos experimentos, resultando em um baixo número de colunas (normalmente, na ordem das unidades ou das dezenas). Assim, concluímos que o exemplo usado aqui é representativo do tipo de problema que estamos procurando enfrentar.

É óbvio que nenhuma das técnicas estatísticas bem estabelecidas, tais como análise de covariância ou outros métodos de regressão, linear ou não linear, seria capaz de extrair relacionamentos significativos a partir de dados tão escassos, posto que elas necessitam de conjuntos de tamanho estatisticamente significativo para trabalhar. Logo, a aplicação de um algoritmo como o proposto aqui, que é capaz de lidar com dados tão esparsos, mostra-se uma alternativa interessante.

Existem três aspectos fundamentais na resposta às reduções de temperatura: aclimatação (aumento na tolerância ao frio), invernalização (redução no tempo de inflorescência) e estratificação (fim da latência das sementes induzida pelo frio). Todos os três são resposta básicas de uma planta ativadas pelas baixas temperaturas e são, em grande parte, determinadas pelas alterações dos padrões de expressões genéticos quando uma planta está submetida a baixas temperaturas (GILMOUR *et al.*, 1998).

Dentro desta resposta, existem alguns genes que a pesquisa identificou como parte dos elementos reguladores fundamentais. Estes genes são 16062_s_at, 17520_s_at e

16111_f_at, enquanto que os genes controlados por estes, nós temos 15611_s_at, 15997_s_at, 13018_at e 13785_at.

Assim, obtemos um problema previamente identificado para que possamos testar a aplicabilidade de nosso algoritmo, que consiste em tentar identificar os genes reguladores dos elementos pertencentes à lista dos controlados acima.

Aplicamos então nosso algoritmo para os dois últimos elementos desta lista (13018_at e 13785_at), conforme descrito nas seções anteriores. Inicialmente, não usamos nenhum tipo de conhecimento prévio que poderia privilegiar os reguladores conhecidos. Usamos os critérios descritos anteriormente para controlar o fim da execução de nosso programa.

Foram testados diferentes números de divisões do intervalo de expressão em conjuntos fuzzy e os resultados semelhantes foram obtidos para o número de conjuntos igual a 3 ou 5, dando-se preferência ao primeiro conjunto dado que por ter menos regras, mostra-se mais simples e compreensível para o usuário que recebe o resultado.

A escolha do número de conjuntos fuzzy por variável, especialmente no caso do elemento cuja regulação está sendo buscada, é uma questão fundamental: aumentar o número de regras também poderia resultar em um casamento mais perfeito do padrão, mas tendo em vista que o resultado busca oferecer hipóteses para uso de um investigador, a simplicidade é um dos objetivos a serem perseguidos. Ademais, sabe-se que o número médio de elementos controladores em células eucarióticas varia de 5 a 6, dependendo do organismo (ARNONE *et al.*, 1997), e nós procuramos fazer com que nossa base de regras contivesse um número próximo destas médias, para ser mais significativo biologicamente.

Para números maiores de conjuntos, percebeu-se uma significativa degradação da desempenho de nosso algoritmo. Esta degradação pode ser devida ao aumento do espaço de busca. Assim, sendo um número maior de gerações e rodadas poderia permitir a obtenção de resultados superiores (que obtivessem uma mlehor identificação do padrão de expressão). Entretanto, os resultados obtidos com um número menor de conjuntos foi satisfatório, chegando a uma aproximação da ordem de 98% do padrão de expressão dos genes em questão. Assim, não prolongou-se esta linha de pesquisa de forma desneessária.

Para o elemento 13875_at o algoritmo evolucionário proposto nesta tese forneceu como resultado o conjunto de regras fuzzy exibido na tabela 5-2 a seguir.

(a) SE Baixo_Nível(17413_s_at) AND NOT Alto_Nível(16111_f_at) ENTÃO Baixo_Nível(13785_at)
(b) SE NOT Médio_Nível (15714_at) ENTÃO Baixo_Nível(13785_at)
(c) SE NOT Médio_Nível (17834_at) ENTÃO Baixo_Nível(13785_at)
(d) SE Baixo_Nível (17421_s_at) ENTÃO Médio_Nível(13785_at)
(e) SE Médio_Nível (16062_s_at) ENTÃO Médio_Nível(13785_at)
(f) SE Baixo_Nível(17050_s_at) OR Alto_Nível(16062_s_at) ENTÃO Alto_Nível(13785_at)
(g) SE Médio_Nível (17034_s_at) ENTÃO Alto_Nível(13785_at)
(h) SE NOT [Alto_Nível(16062_s_at) OR Baixo_Nível(15140_s_at)] ENTÃO Alto_Nível(13785_at)

Tabela 5-2: Conjunto completo de regras obtidas para o gene 13875_at. Dividiu-se o universo de discurso desta variável em 3 conjuntos fuzzy e definiu-se o número médio de regras por conjunto em 2. O operadores de mutação de inserção e de exclusão alteram este número e o resultado foi um conjunto com um total de 8 regras.

Os resultados descritos acima são interessantes, apresentando as seguintes características notáveis:

- As regras (a), (e) e (f) apresentam relacionamentos com reguladores conhecidos.
- A regra (g) apresenta um novo regulador (17034_s_at) que foi considerado interessante por pesquisadores ligados ao estudo da Arabidopsis thaliana para que novos estudos fossem realizados em um futuro próximo.
- As regras (f) e (h) mostram a presença dos elementos 15140_s_at e 17050_s_at que possuem alta correlação com o elemento 17520_s_at, elemento este que, apesar de ser um regulador conhecido, não estava presente em nossas regras. Este tipo de problema deve ser esperado devido ao pequeno número de pontos conhecidos, o que leva a qualquer tipo de afirmação sobre a presença destes genes na base de regras de regulação ser prematura.

Depois do sucesso desta investida inicial nós realizamos uma busca usando o nosso algoritmo para uma regulação para o gene 13018_at, outro que está envolvido na resposta ao frio. Desta vez, para testar mais nosso programa, nós pedimos para que fosse

necessariamente incluída uma ativação por parte do gene 17520_s_at e, preferencialmente, uma inibição por parte do gene 16111_s_at. Isto serve como teste para o mecanismo de inserção de conhecimento prévio.

Um relacionamento de ativação de um gene X para um gene Y pode ser inserido pedindo-se para o programa inserir o conjunto fuzzy *Alto(X)* como pré-condição de uma das regras escolhidas para o conjunto *Alto(Y)*, enquanto que uma inibição pode ser coseguida através da inserção do mesmo conjunto fuzzy *Alto(X)* como pré-condição do conjunto *Baixo(Y)*.

O programa permite que sejam incluídos relacionamentos mandatórios (que têm que estar presentes no conjunto final de regras), como foi feito no caso do gene 17520_s_at, ou preferenciais (cuja presença no conjunto final de regras é desejável, mas não obrigatória), como no caso do gene 16111_s_at. Estes dois genes reguladores foram escolhidos de forma aleatória entre os genes reguladores conhecidos. Usando o mesmo tipo de algoritmo descrito para o elemento 13875_at, as regras obtidas são aquelas descritas na tabela 5-3 a seguir.

(a) SE Baixo_Nível(17520_s_at) OR Alto_Nível(20351_at) ENTÃO Baixo_Nível(13018_at)
(b) SE NOT [Médio_Nível(14969_at) OR Baixo_Nível(17034_s_at)] ENTÃO Baixo_Nível(13018_at)
(c) SE Médio_Nível (13018_at) ENTÃO Lowly_Medium(13018_at)
(d) SE Baixo_Nível (17421_s_at) AND Alto_Nível(16218_s_at) ENTÃO Médio_Nível(13018_at)
(e) SE Médio_Nível (1611_s_at) AND Médio_Nível(17200_at) ENTÃO Médio_Nível(13018_at)
(f) SE Baixo_Nível(17034_s_at) AND NOT Alto_Nível(14832_at) ENTÃO Alto_Nível(13018_at)
(g) SE Alto_Nível (17520_s_at) ENTÃO Alto_Nível(13018_at)

Tabela 5-3: Conjunto de regras obtidas para o gene 13018_at.

Os resultados obtidos acima acusam a presença do relacionamento requerido com o gene 17520_s_at nas regras (a) e (g). O relacionamento preferencialmente escolhido, com o gene 16111_s_at também está presente na regra (e). O gene 17034_s_at, considerado como um achado promissor nas regras anteriores, foi novamente encontrado neste conjunto de regras.

5.5.2 Sistema nervoso central de ratos

O segundo conjunto de dados utilizados para testar nosso programa foi aquele obtido pela expressão da rede de regulação genética do sistema nervoso central (SNC) de um rato durante seus estágios embrionário e de recém nascido.

A Carboxilase do Ácido Glutâmico (GAD) que é a enzima responsável pela conversão do ácido glutâmico em ácido gama-aminobutírico (GABA), sustância esta que é o principal transmissor inibidor nas regiões superiores do cérebro e um hormônio putativo no pâncreas. As duas formas molecular do GAD (respectivamente com 65kD e 67kD), que têm 64% de aminoácidos idênticos, são altamente conservadas e expressas no SNC, pâncreas, testes e ovário, tendo um importante papel no desenvolvimento do SNC.

D'HAESELEER *et al.* (2000) mostra os relacionamentos mais prováveis dos elementos GAD65 e GAD67 em um diagrama que podemos ver na figura 5-9. Usando estes relacionamentos como objetivo máximo, nós aplicamos o nosso algoritmo para a determinação da rede de regulação do elemento preGAD67. Este elemento, além de ser instrumental na formação do elemento GAD67, é altamente conectado, tendo 5 entradas teorizadas, o que faz dele um candidato interessante para nosso algoritmo.

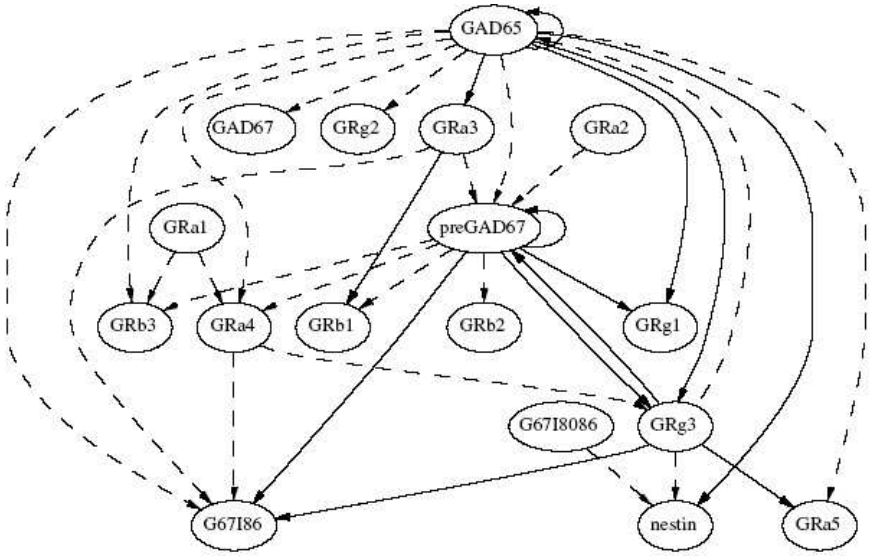


Fig 5-9 → O diagrama de interação genética, conforme mostrado em (D'HAESELEER et al., 2000).As linhas pontilhadas indicam inibição, enquanto que as linhas inteiriças indicam a existência de um relacionamento de ativação entre o gene de origem da seta e o gene de destino desta.

Executando o programa implementado para esta tese da mesma forma que o descrito na seção anterior, foi obtido o conjunto de regras descrito na tabela 5-4, que é representado graficamente pela figura 5-10.

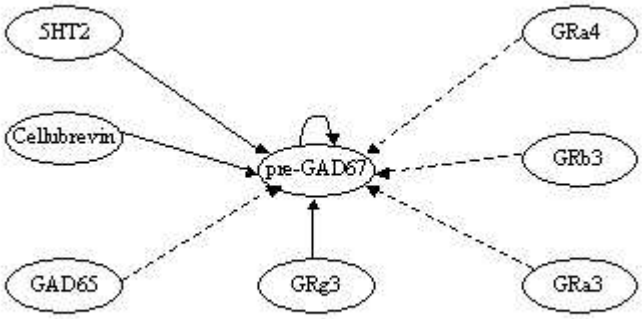


Figura 5-10 → Representação gráfica das regras descritas pelo conjunto de regras da tabela 5-4. As setas seguem o padrão descrito na figura 5-9 (linhas inteiras para ativação, linhas tracejadas para inibição). Os relacionamentos foram extraídos baseado no pressuposto de que se um alto nível do gene X implica em um baixo nível de pre-GAD67, então o relacionamento é de inibição. Todos os outros relacionamentos foram considerados como sendo de ativação.

(a) SE Baixo_Nível(5HT2) AND Médio_Nível(cellubrevin) ENTÃO Baixo_Nível(pre-GAD67)
(b) SE Alto_Nível (GAD65) ENTÃO Baixo_Nível(pre-GAD67)
(c) SE Baixo_Nível (pre-GAD67) ENTÃO Baixo_Nível(pre-GAD67)
(d) SE Baixo_Nível (GRg3) ENTÃO Baixo_Nível(pre-GAD67)
(e) SE Médio_Nível (GRa3) ENTÃO Médio_Nível(pre-GAD67)
(f) SE Alto_Nível (GAD65) ENTÃO Médio_Nível(pre-GAD67)
(g) SE Baixo_Nível (GRb3) ENTÃO Alto_Nível(pre-GAD67)
(h) SE Baixo_Nível (GRa4) ENTÃO Alto_Nível(pre-GAD67)
(i) SE Baixo_Nível (GRa3) ENTÃO Alto_Nível(pre-GAD67)

Tabela 5-4: Conjunto de regras obtidas para o gene pre-GAD67

Estas regras apresentam alguns resultados notáveis, descritos a seguir:

1. O GAD65, que é colocado como um repressor, aparece em duas regras na base acima: primeiro na regra (b), na qual sua alta expressão induz uma baixa expressão de pre-GAD67, o que é um forte comportamento inibitório, e na regra (f) na qual encontra-se que sua alta expressão induz um médio nível de expressão de pre-GAD67, o que é um comportamento levemente inibitório. É óbvio que esta segunda regra poderia ser encarada também como uma leve forma de ativação, mas, combinada com a regra anterior, leva a um comportamento fortemente inibitório.
2. Outro repressor putativo, GRa3, aparece com um relacionamento semelhante ao do GAD65, como podemos ver nas regras (e) e (i). Nesta última regra temos um claro exemplo de comportamento inibidor, pois a ausência de Gra3 (indicada pela pré-condição de existência de um baixo nível de expressão deste gene) implica na forte expressão do gene de interesse.
3. O pre-GAD67 tem um mecanismo putativo de ativação que foi descoberto pela regra (c). Entretanto, usar esta regra como um sucesso de nosso algoritmo é algo temerário, visto que um elemento sempre demonstra uma alta correlação consigo próprio. Assim, regras que incluam o próprio elemento, especialmente como a regra (c), que prediz um baixo nível de expressão a partir de outrobaixo nível de expressão, aparecerão de forma

freqüente e a não ser que o conhecimento prévio diga o contrário, devem ser desconsideradas.

4. A regra (d) mostra outro relacionamento putativo dscoberto pelo nosso programa, através da correta previsão de um relacionamento de ativação por parte do gene GRg3. Um baixo nível de um ativador necessário implicará em um baixo nível de expressão do gene de interesse na maioria das situações. Isto só não é totalmente verdade quando existe um relacionamento sigmoidal ou exponencial entre o gene regulado e o regulador, quando a presença de poucas moléculas deste último causarão a forte expressão do primeiro. Entretanto, pela natureza de nosso algoritmo, e pelo fato de nós desprezarmos os elementos com baixíssimos nívis de expressão (como descrito anteriormente na seção sobre pré-processamento de dados), este tipo de relacionamento não é captado pelo nosso algoritmo.
5. O gene GRa4 é inibido pelo preGAD67. Entretanto, apesar do algoritmo captar o relacionamento de inibição entre os dois genes, este foi invertido. Isto provavelmente poderia ser remediado com mais dados.

Estes resultados são interessantes, pois dos sete candidatos presentes na base de regras retornada por nosso algoritmo (5HT2, cellubrevin, GAD65, pre-GAD67, GRg3, GRa3, GRb3 e GRa4), quatro deles estão presentes na lista de reguladres putativos do preGAD67.

Para demonstrar a capacidade de modelagem das regras expostas anteriormente, são mostradas na figura 5-11 as trajetórias reais e calculadas do GAD67. Apesar de haver algumas diferenças de níveis de expressão a forma das trajetórias é seguida de forma precisa. Isto sugere que a regulação tenha sido capturado de forma correta e que o processo subjacente aos valores de expressão tenha sido modelado de forma razoavelmente precisa.

Assim, pode-se concluir que estas bases de regras são candidatas promissoras a maiores avaliações, devendo ser testadas contra novas bases de dados e eventualmente levadas às bancadas de laboratórios, para a realização de experimentos mais específicos que possam validar seu valor.

Os resultados individuais de cada regra foram mostrados na figura 4-8 como demonstração dos critérios de parada usados pelo algoritmo e merecem uma discussão

especial. O melhor resultado obtido obteve uma avaliação de 105, que corresponde a um erro de aproximadamente 1% em cada ponto da trajetória.

A pior das rodadas obteve avaliação igual a 21, o que corresponde a um erro de aproximadamente 5% em cada ponto da trajetória. A média global das avaliações ficou em 58,5, com desvio padrão de 28,9. Esta alta dispersão confirma o que foi discutido na seção introdutória sobre algoritmos genéticos: não é razoável esperar que todas as rodadas obtenham um bom resultado. Ademais, tendo em vista o uso dos melhores indivíduos de cada rodada como inicialização da última rodada, é de se esperar que esta tenha uma avaliação superior à melhor de todas as outras rodadas.

Validar o algoritmo genético pela média de suas rodadas é similar ao conceito de validação do método de Newton-Raphson com inicialização aleatória aplicado a uma função multi-modal (RUGGIERO *et al.*, 1996). Em execuções distintas, como inicialização em diferentes pontos do espaço de busca, este método encontrará máximos distintos (locais e/ou global). Entretanto, o resultado final não consiste em calcular a média dos máximos encontrados, mas sim retornar o maior dos máximos encontrados. Pode-se traçar um paralelo entre o paradigma de utilização deste algoritmo e o padradigma dos algoritmos evolucionários - o fato de que haverá uma dispersão dos resultados de várias rodadas é uma característica inerente ao uso destes que deve ser conhecida *a priori* aqueles que desejem usá-los e não algo que os desabone como método.

5.6 Comentários gerais sobre a aplicação de engenharia reversa

Este modelo apresenta resultados interessantes quando lidando com dados esparsos como os provenientes de microarrays. Os resultados mostrados nesta tese assemelham-se sobremaneira àqueles conhecidos pelo biólogo e existe uma perspectiva interessante para

que o programa criado para esta tese transforme-se em uma ferramenta verdadeiramente útil para que biólogos gerem hipóteses testáveis em bancada.

Os relacionamentos regulatórios encontrados neste trabalho provavelmente não seriam descobertos por outros modelos, devido ao grande número de genes e o pequeno número de pontos, que inviabiliza métodos estatísticos tradicionais, além de outros métodos que requerem uma grande quantidade de dados, como redes neurais.

Entretanto, quando houver pontos suficientes para tanto, acreditamos que métodos estatísticos estabelecidos ou quaisquer outros dependentes de grande quantidade de dados devem ter a preferência sobre nosso algoritmo, que na verdade é uma heurística e, como tal, tem um grau de confiabilidade menor.

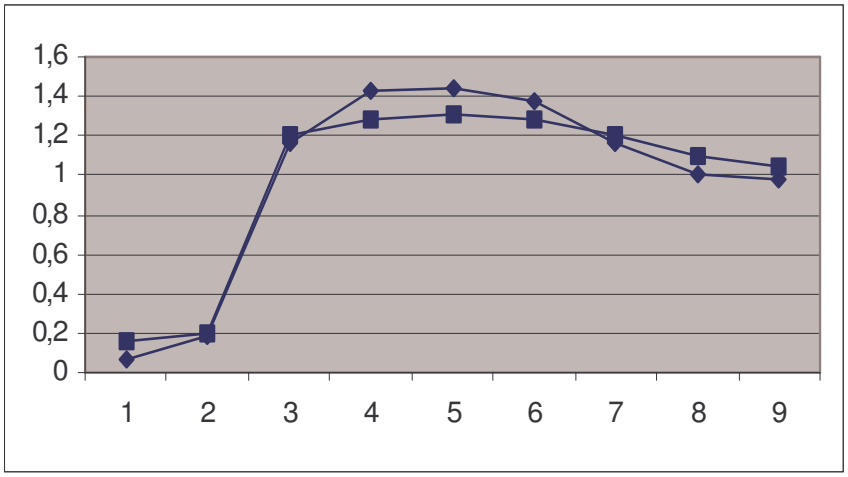


Fig 5-11 → Trajetórias calculada (quadrados) e real (losângulos) para o preGAD67 no sistema nervosa central de ratos em estágios iniciais de desenvolvimento. No eixo das abcissas, têm-se os instantes de tempo em que as medidas foram efetuadas, enquanto que no eixo das coordenadas estão os valores de expressão do gene estudado. As curvas são semelhantes e mesmo nos pontos de divergência o formato das curvas é similar.

Ademais, posto que nossas conclusões são baseadas em uma quantidade de dados que não é estatisticamente significativa, é possível, e até provável, que nosso método obtenha em várias situações resultados espúrios, especialmente falsos positivos. Por conseguinte, deve ficar claro que o objetivo do nosso algoritmo consiste em encontrar candidatos interessantes para verificação posterior, e não resultados finais mostrando a regulação real de certos genes de interesse. É óbvio que a intervenção de um investigador qualificado sempre se mostrará necessária ao fim da execução de nosso programa.

Outro ponto a ser levado em consideração é que, como mencionamos anteriormente, métodos dependentes apenas de dados não são a solução final para nenhuma aplicação em

bioinformática. Deve-se sempre procurar incluir o máximo possível do conhecimento disponível na biologia para que nossos modelos sejam mais completos.

Um exemplo desta questão é o ponto onde escolhemos os candidatos a reguladores que serão testados. Neste trabalho nós usamos apenas os dados (somente aqueles que têm níveis de expressão significativos) e conhecimento prévio das regulações existentes, no caso da *Arabidopsis thaliana*. Entretanto, existem maneiras mais interessantes de se fazer esta escolha que mereceriam um estudo posterior.

Uma delas seria deixar como candidato a regulador somente aqueles elementos cujas seqüências apresentassem um domínio de ligação. Domínios de ligação são elementos fortemente estudados na área da biologia e correspondem a seqüências de aminoácidos que têm características físicas especiais que permitem que as proteínas se liguem ao DNA, passo primordial para que a regulação tenha efeito. Assim, se um elemento não possui um domínio de ligação podemos afirmar com certeza que ele não tem função regulatória.

Existem vários trabalhos que mostram algoritmos computacionais para o reconhecimento de domínios de ligação. DODD *et al.* (1990) e PIETROKOVSKI *et al.* (1997) descrevem técnicas algorítmicas para o reconhecimento de domínios helix-turn-helix e o trabalho descrito em (OHLER, 2002). Embutir eses trabalhos dentro de nosso algoritmo significaria uma pré-seleção mais eficiente que seria tão eficiente quanto o método atual em termos de diminuição da dimensão do espaço de busca mas com a presença apenas de elementos que são candidatos a reguladores mais prováveis.

O único problema deste ganho é que a descoberta de promotores é um ramo de pesquisa por si só, o que é claramente representado pelo fato de um dos trabalhos que citamos acima ser uma tese de doutorado. Além dos trabalhos já citados, existem vários outros já feitos, mas a análise de todos os trabalhos ainda não é 100% precisa, o que efz com que se optasse pela sua não aplicação neste instante.

Além da complexa biologia dos promotores merecer ampla discussão, omitida aqui, existem vários fatores estocásticos ou não que implicam no uso de técnicas tais como Cadeias de Markov, Redes Bayesianas ou Redes Neurais para a realização deste trabalho, e nenhuma delas consegue 100% de precisão na descoberta de promotores. É óbvio que o fato de tal avanço ser conseguido apenas a custa de muitos esforços não desmerece a sua aplicação, mas implica sim na impossibilidade de sua aplicação no âmbito deste trabalho.

Outro avanço interessante quando lidando com o problema de regulação celular consiste em que se possa lidar com as questões dos diferentes atrasos, e tempos de degradação dos diferentes reguladores, além da questão da assincronia da atuação entre os mesmos, mas nosso modelo ainda não é capaz de fazê-lo. Isto afeta sobremaneira o resultado do sistema. Entretanto, nosso algoritmo só é capaz atualmente de realizar alterações síncronas o que afeta a sua capacidade de modelagem. Apesar dos resultados obtidos com este modelo simplificado serem já interessantes, pode-se considerar o desenvolvimento de modelos futuros (e mais analíticos) deste problema que possam tentar resolver este tipo de questão.

Capítulo 6 - Conclusão e Trabalhos Futuros

Neste capítulo serão discutidos os resultados obtidos nas duas áreas de aplicação do algoritmo proposto nesta tese, incluindo suas qualidades e limitações. Ao final desta discussão, serão apresentadas algumas idéias de como o trabalho desenvolvido aqui poderia ser incrementado de forma que os resultados obtidos melhorem e se tornem mais significativos.

6.1 Algoritmo Proposto

Esta tese apresentou e estudou o problema de obtenção de conhecimento de fontes escassas de dados que possivelmente incorporam valores numéricos e categóricos, de forma que este conhecimento seja claro e compreensível para o usuário final da informação.

Apresentou-se um algoritmo evolucionário para o desenvolvimento de bases de regras fuzzy que foram aplicados em uma série de problemas importantes que não possuem boas soluções na literatura.

O algoritmo apresentado aqui possui um tempo de execução longo, que foi compensado pela qualidade dos resultados obtidos, tanto em termos de precisão como de compreensibilidade dos resultados.

Esta deriva do fato de que a lógica fuzzy trabalha com termos lingüísticos que são usados no discurso cotidiano. Em outras palavras, o formato de regras fuzzy permite que os resultados sejam compreendidos mais facilmente em comparação com outros métodos amplamente difundidos, tais como máquinas de vetor de suporte (ONODA *et al.*, 2001; AUER *et al*, 2002), métodos de clustering (KAWAJI *et al.*, 2001; Hanisch *et al.*, 2002) e métodos matriciais (VAN SOMEREN *et al.*, 2000), cujos resultados apresentam um bom grau de precisão, mas que, do ponto de vista da pessoa que recebe seus resultados (o usuário final dos dados), são caixas-pretas.

Em contrapartida, o método aqui proposto apresenta seus resultados em um formato de regras que segue o padrão lingüístico usualmente adotado pelas pessoas, o que facilita sua compreensão e a sua usabilidade posterior.

Outro ponto importante apresentado aqui foi a possibilidade demonstrada neste trabalho de incorporar conhecimento pré-existente para diminuir o espaço de busca do algoritmo.

O uso de regras fuzzy tem uma vantagem inerente sobre os modelos baseados em equações diferenciais, tais como apresentados em (CHEN *et al.*, 1999; SAKAMOTO *et al.*, 2001), por exemplo. Estes últimos são dependentes de parâmetros específicos e obtêm resultados que podem divergir da realidade se estes parâmetros tiverem um pequeno erro. A lógica fuzzy é menos sujeita a estes erros, tendo em vista que pequenas mudanças em parâmetros numéricos, tais como os conjuntos fuzzy, não causam alterações bruscas nos resultados obtidos.

Esta propriedade também é denominada robustez e é uma característica inerente dos algoritmos fuzzy, ao passo que é necessário provar a robustez de outros tipos de algoritmos. Pequenas variações em parâmetros de modelos matriciais ou diferenciais causam a modificação das matrizes básicas do sistema, podendo alterar dramaticamente sua dinâmica. No caso fuzzy, tendo em vista sua estrutura pseudo-lingüística, pequenas alterações em parâmetros como conjuntos ou nas condições iniciais do sistema vão causar

perturbações de magnitude semelhante no desenvolvimento do mesmo, sem alterações consideráveis na sua dinâmica.

Outra vantagem do modelo aqui utilizado é que ele pode incorporar diretamente dados Booleanos, analisando-os sem dificuldades adicionais. Isto se torna mais difícil para os modelos diferenciais uma vez que dados Booleanos são inerentemente discretos e como tal não possuem derivadas que possam ser incorporadas aos modelos envolvendo equações diferenciais contínuas.

6.2 Aplicações em Bioinformática

As aplicações biológicas já mencionadas são os resultados mais significativos apresentados nesta tese. Usando apenas a implementação do algoritmo apresentado e um conhecimento pré-existente da biologia, representado em forma de regras de regulação, conseguiu-se descobrir, entre um pequeno número de candidatos a regulador, aqueles genes que pertencem ao processo regulatório de genes de interesse, fato confirmado por experimentos biológicos (GILMOUR *et al.*, 1998).

Os resultados obtidos aqui devem ser compreendidos como uma demonstração de que é possível orientar a pesquisa para determinados candidatos mais promissores, evitando uma busca cega e dispendiosa.

Entretanto, posto que os resultados são baseados em uma quantidade de dados pequena e não estatisticamente relevante, é possível que alguns resultados espúrios tenham sido obtidos, especialmente em termos de falsos positivos. Dado o tamanho dos conjuntos de dados com que se trabalhou, qualquer método teria este problema, que tende a diminuir com o aumento do número de trajetórias.

Mesmo levando em consideração estas limitações inerentes aos conjuntos de dados utilizados, é importante ressaltar que os resultados obtidos apontam para uma eficiente descoberta de candidatos a reguladores, posto que até 99,9% dos genes não reguladores foram eliminados do conjunto de resultados, no caso da análise dos dados da *Arabidopsis thaliana*.

Ademais, dentro dos resultados obtidos encontram-se aqueles genes pelos quais se estava buscando inicialmente, isto é, aqueles genes cujo envolvimento na resposta ao frio é conhecida de antemão (GILMOUR *et al.*, 1998). Isto significa que com a execução do algoritmo apresentado, utilizando os dados de apenas um experimento de microarray, pôde-se obter resultados que, se transplantados para um laboratório de biologia, poderiam guiar uma pesquisa de forma que os microarrays realizados a seguir testassem os relacionamentos mais promissores, o que permitiria que se obtivesse resultados mais rapidamente e com um menor ônus financeiro.

No caso do sistema nervoso central dos ratos, a comparação com um trabalho realizado anteriormente (D'HAESELEER *et al.*, 2000) e geralmente aceito como biologicamente plausível aponta para uma grande similaridade com os resultados conseguidos através da aplicação do algoritmo proposto nesta tese.

Ademais, a captura bastante precisa das variações nos níveis de expressão dos genes estudados sugere que o processo regulatório que gera estas variações pode ter sido modelado de forma correta, fazendo por justificar a realização de testes de bancada para uma posterior comprovação.

Em ambos os casos, junto com os reguladores previamente identificados, existem vários outros candidatos, alguns dos quais são provavelmente espúrios (falsos positivos). Isto é esperado, dado o pequeno volume de dados disponível em cada caso. Entretanto, o algoritmo ainda se justifica, posto que sua proposta original era restringir o espaço de busca para o trabalho de bancada em laboratórios biológicos.

Deve-se manter sempre em mente que as bases de regras obtidas através da aplicação do método aqui proposto devem ser tratadas apenas como candidatos interessantes para o problema sendo modelado, e não como um resultado final. Assim, mais testes, especialmente na bancada de um laboratório de biologia, serão necessários para futura validação.

A incorporação de informação pré-existente é um conceito lógico que tem sido pouco utilizado na literatura. Por exemplo, no caso dos exemplos biológicos apresentados no capítulo 5 desta tese, existe uma ampla gama de relacionamentos regulatórios entre genes que são conhecidos atualmente. Ignorar estes conhecimentos significa ampliar de

forma desnecessária o espaço de busca do algoritmo, arriscando-se a obter resultados menos precisos ou simplesmente incorretos.

Outro tipo de conhecimento utilizado nesta aplicação foi a seleção dos elementos que poderiam ser os reguladores putativos dos genes de interesse, que limitou fortemente o espaço de busca. Uma forma mais inteligente de selecionar os reguladores é discutida na seção de trabalhos futuros, a seguir.

Não é razoável procurar uma solução com todos os genes de um microarray apenas porque eles estão disponíveis. O uso de conhecimento prévio pode aperfeiçoar os resultados obtidos e minimizar a necessidade de dados.

Assim, torna-se imperativo incluir nos modelos de busca o atual estado da arte do conhecimento no campo biológico, de forma que o modelo obtido seja o mais completo possível (SCHRAGER *et al.*, 2002).

O algoritmo proposto para esta tese permite a introdução deste conhecimento de forma a restringir o espaço de busca e fazer com que os resultados obtidos sejam o mais próximo possível da realidade.

6.3 Trabalhos Futuros

Nesta seção descreveremos algumas melhorias que podem ser feitas no algoritmo apresentado nesta tese, de forma a obter resultados mais significativos e/ou fazê-lo mais capaz de lidar com problemas tais como aqueles aqui apresentados.

Os trabalhos apresentados a seguir não foram implementados devido ao fato de serem complexas adições que merecem estudos intensivos, não sendo apenas pequenas adições ao código fonte do sistema desenvolvido para esta tese ou estudos que se possam realizar em um curto espaço de tempo.

6.3.1 Seleção de candidatos a reguladores

O processo de regulação da transcrição de um gene é um processo complexo que normalmente envolve a ligação de uma série de proteínas reguladoras de transcrição (fatores de transcrição) a regiões especiais (sítios de ligação, ou *binding sites*) situadas um pouco antes (*upstream*) da seqüência de DNA codificadora do gene sendo regulado.

A localização dos sítios de ligação é outro fator complexo. Existem promotores que podem estimular a transcrição ligando-se a um promotor localizado a dezenas de kilobases de distância da seqüência codificadora, como, por exemplo, o promotor SV40, enquanto que outros somente funcionam de forma proximal, distantes no máximo 15-20 bases da seqüência do gene, como os elementos controlando o gene HSV tk. Existe, obviamente, um grande número de genes localizados entre os dois extremos (ALBERTS *et al.*, 2002).

Os fatores de transcrição são compostos por duas regiões funcionais essenciais: um domínio de ligação ao DNA e um domínio de ativação. O domínio de ativação ao DNA consiste de uma seqüência de aminoácidos específicos que reconhecem os sítios de ligação com os quais devem se ligar (OHLER, 2001).

Os fatores de transcrição são tipicamente classificados de acordo com a estrutura do seu domínio de ligação que pode ser de um dos seguintes grupos:

- Zinc fingers
- Helix-Turn-Helix
- Zíper de Leucina
- Helix-Loop-Helix
- Grupos de alta mobilidade

Por outro lado, o domínio de ativação interagem com os componentes do aparato de transcrição, especialmente a RNA polimerase e com outras proteínas regulatórios afetando a eficiência da transcrição do DNA. Estes fatores podem ser ativados por estímulos fisiológicos, terapêuticos e patológicos.

A existência e especificidade dos domínios de ligação pode ser explorada na seleção de candidatos a reguladores que seriam submetidas ao algoritmo proposto nesta tese. Em

vez de selecionar os candidatos pelo grupo funcional, como feito aqui, poderia se analisar a sequência codificadora de cada proteína expressa na célula e permitir que o algoritmo utilizasse como candidato somente aqueles que possuíssem um domínio de ligação reconhecível.

Este trabalho não foi realizado nesta tese pois a complexidade do assunto é imensa. O reconhecimento de domínios de ligação é uma área de pesquisa avançada que tem suscitado o desenvolvimento de várias teses de doutorado em todo o mundo. Entretanto, os resultados destes trabalhos poderiam ser usados como forma de fazer com que os candidatos a regulador fossem somente aqueles capazes de realizar tal função, diminuindo o número de resultados espúrios obtidos pelo algoritmo.

6.3.2 Integração com um laboratório de bioinformática

Os resultados obtidos no trabalho realizado em bioinformática nesta tese foram promissores. Por exemplo, um pesquisador que recebesse como resultado o conjunto de regras para o gene 13875_at, por exemplo, teria uma lista de 9 genes que são considerados como candidatos promissores pelo algoritmo proposto nesta tese.

Estes genes foram retirados de um espaço de 8000 genes, o que implica em um fator de redução do espaço de busca de cerca de 99,9% pela aplicação de um algoritmo computacional, o que pode ser de grande valia para um pesquisador trabalhando na área biológica.

É razoável assumir que dentre os genes desprezados existem um certo número falsos negativos, mas neste instante é impossível obter uma estimativa precisa deste número. Ademais, este número variará para cada aplicação, dependendo de fatores como nível de expressão dos genes envolvidos em relacionamentos regulatórios, por exemplo. Isto quer dizer que se um gene regulatório não estiver expresso de forma expressiva dentro do conjunto de dados, ele pode vir a ser eliminado pelos passos de pré-processamento e não ser considerado pelo algoritmo.

Este e outros resultados obtidos nesta tese sugerem que o algoritmo aqui proposto pode ser uma ferramenta útil para ajudar a descobrir processos regulatórios ainda desconhecidos de forma que novas hipóteses possam ser geradas e testadas em um laboratório (*wet lab*).

Para tanto, seria interessante que se desenvolvesse uma cooperação com um laboratório de pesquisas biológicas, de forma que o algoritmo pudesse ser usado de forma extensiva e suas capacidades testadas em situações de necessidades reais. Assim, suas verdadeiras capacidades e limitações poderiam ser testadas e novos aprimoramentos propostos.

Bibliografia

ALBA, E., CHICANO, J. F., 2004, "Training Neural Networks with GA Hybrid Algorithms", In: *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2004)*, pp. 864-876, Seattle, EUA, Jun.

ALBERTS, B., JOHNSON, A., LEWIS, J., *et al.*, 2002, "*Molecular Biology of the Cell*", 4ª. Edição, Boston, Garland Publishing.

ALTER, O. , BROWN, P. O., BOTSTEIN, D., 2001, "Processing and modeling genome-wide expression data using singular value decomposition", In: Bittner M. L., Chen Y. (eds) *Microarrays: Optical Technologies and Informatics*, cap 12., pp. 171-186, International Society for Optical Engineering, Bellingham, EUA

AKUTSU, T., MIYANO, S., KUHARA, S., 1999 "Identification Of Genetic Networks From A Small Number of Gene Expression Patterns Under The Boolean Network Model", In: *Proceedings of the Pacific Simposium on Bio-Informatics (PSB'99)*, pp. 17-28, Havaí, EUA, Jan.

ANDO, S; IBA H., 2000, "Identifying the Gene Regulatory Network by Real-Coded, Variable-Length, and Multiple-Stage GA",In: *Proceedings of 2001 Congress on Evolutionary Computation*, pp. 712-719, Seul, Coréia do Sul, Maio.

ARNONE, M.I., DAVIDSON, E.H., 1997, "The hardwiring of development: organization and function of genomic regulatory systems", *Development* v. 124, n. 10 (May), pp. 1851-1864.

AUER, P., BURGSTEINER, H., AND MAASS,W., 2002, "Reducing communication for distributed learning in neural networks". *International Conference on Artificial Neural Networks (ICANN 2002),* pp. 123-128, Madri, Espanha, Aug.

BAESENS, B.; EGMONT-PETERSEN, M.; CASTELO, R. *et al.*, 2002, "Learning Bayesian network classifiers for credit scoring using Markov Chain Monte Carlo search", *Proceedings of the 16th International Conference on Pattern Recognition (ICPR)*, pp. 49-52, Quebéc, Canadá, Jul.

BARANAUSKAS, J. A.., MONARD, M. C., 2003, "Conceitos sobre aprendizagem de máquina", In: Rezende, S. O. (coord), "*Sistemas Inteligentes – Fundamentos e Aplicações*", 1ª. Edição, capítulo 4, São Paulo, Brasil, Ed. Manole

BATISTA, G. E. A. P. A., PRATI, R. C., MONARD, M.C., 2004, "A Study of the Behaviour of Several Methods for Balancing Machine Learning Training Data", *SIGKDD Explorations*, v. 6, n. 1 (Dec), pp. 20-29.

BENTLEY, P., 1999, "Evolving Fuzzy Detective: An Investigation in the Evolution of Fuzzy Rules", In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '99)* , pp. 38-47, Orlando, EUA, Jul.

BLAVYAS, I., KIMMEL, R., 2001, *Efficient Classification via Multiresolution Training Set Approximation*, Technical Report 3, Technion, Haifa, Israel.

CANTÚ-PAZ, E., 1997, *A Survey of Parallel Genetic Algorithms*, Technical Report, University of Illinois, Chicago, EUA.

CELIS, J. E.; KRUHOFFER, M., 2000, "Gene expression profiling: monitoring transcription and translation products using DNA microarrays and proteomics", *FEBS Letters* n. 480, pp. 2-16

CHEN, T., 1999, "Modeling Gene Expression With Differential Equations", In: *Proceedings of the Pacific Simposium on Bio-Informatics (PSB'99)*, pp. 29-40, Havaí, EUA, Jan.

CHEN, K., LIU L., 2003, *Towards Finding Optimal Partitions of Categorical Datasets*, Technical Report, Syracuse University, EUA

CHO, R. J.; HUANG, M.; DONG, H., *et al*., 2001, "Transcriptional Regulation and Function in the Human Cell Cycle" Nature *Genetics*, v. 27, n. 1 (Jan), pp. 48-54.

COOPER, S. ; SHEDDEN, K., 2003, "Microarray analysis of gene expression during the cell cycle", *Cell & Chromosome*, v. 2, n. 1, pp. 66-70, BioMed Central Ltd.

CLARK, P., NIBLETT, T., 1987, "Induction in Noisy Domains" In *Proceedings of the 2nd European Working Session on Learning*, pp. 11-30, Bled, Yugoslavia, Jul.

CRISTOFOR, D. AND SIMOVICI, D. A., 2002, "An information theoretical approach to clustering categorical databases using genetic algorithms", *Second SIAM International Conference on Data Mining, Workshop on clustering high dimensional data*, n. 3, Arlington, EUA, Apr.

DASGUPTA, D., GOMEZ, J., 2002, "Evolving Fuzzy Classifiers for Intrusion Detection", Proceedings of the 2002 IEEE Workshop on Information Assurance, pp. 118-129, Nova York, EUA, Jun.

DAVIDSON, E., RAST, J. P., 2002, "A Genomic Regulatory Network for Development", *Science Magazine*, v. 295, n. 10 (Mar), pp. 1670-1679.

DE JONG, H., 2002, "Modeling and Simulation of Genetic Regulatory Systems : A Literature Review", *Journal Of Computational Biology*, v. 9, n. 1 (Jan), pp. 67-103, Ed. MaryAnnLiebert,Inc.

DELLAERT. G., 1995, *Towards a biologically defensible model of development*, MSc Thesis, Case Western Reserve University, EUA.

D'HAESELEER, P., 1993, *Reconstructing Gene Networks from Large Scale Expression Data*, Tese de Doutorado, The University of New Mexico, Novo México, EUA

D'HAESELEER, P; LIANG, S., SOMOGYI, R., 2000, "Genetic Network Inference – from Co-Expression Clustering to Reverse Engineering", *Bioinformatics* v. 16, n. 8 (Aug), pp. 707-726, Oxford University Press.

DOUNIAS, G.; TSAKONAS, A., JANTZEN, J. *et al.*, 2002 "Genetic Programming for the Generation of Crisp and Fuzzy Rule Bases in Classification and Diagnosis of Medical Data", *Proc. Of the Neuro-Fuzzy Conference (NF' 2002)*, pp. 60-67, Havana, Cuba, Jan.

DODD, I. B., EGAN J.B., "Improved detection of helix-turn-helix DNA-binding motifs in protein sequences", *Nucleic Acids Res* v. 18 n. 26 (Jun) pp. 5019-5026

DORIGO, M., MANIEZZO, V., 1992, "Parallel Genetic Algoriths : Introduction and overview of current research", In J.Stenders (ed.), "Parallel Genetic Algorithms: Theory and Applications", IOS Press, pp. 5-42.

DUCH, W.; ADAMCZAK, R., GRABCZEWSKI, K., 2001 "A New Methodology Of Extraction, Optimization And Application Of Crisp And Fuzzy Logical Rules", *IEEE Transactions On Neural Networks*, v. 12, N. 2 (Mar), pp. 277-306.

DUGGAN, D. J.; BITTNER, M.; CHEN, Y.; et al., 1999, "Expression profiling using cDNA microarrays", Nature Genetics Suppl., vol. 21, N. 1 (Jan), pp. 10-14

EGGERMONT, J.; KOK, J. N. E KOSTERS, W. A., 2004a, "Genetic Programming for Data Classification:Partitioning the Search Space", *Proceedings of the 19th Annual ACM Symposium on Applied Computing (SAC'04)*, pp. 1001-1005 Nicosia, Chipre, Mar.

EGGERMONT, J.; KOK, J. N., KOSTERS, W. A., 2004b, "Detecting and Pruning Introns for Faster Decision Tree Evolution", In: *Proceedings of the 8th International Conference on Parallel Problem Solving From Nature*, pp. 1068-1077, Birmmingham, Sep.

ELKAN, C., 2001, "The Foundations of Cost-Sensitive Learning", In: *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI'01)*, pp. 617-628, Seattle, EUA, Aug.

ESPINDOLA, R. P., 2004, *Sistemas Inteligentes para Classificação de Dados*, Tese de Doutorado, COPPE-UFRJ, Rio de Janeiro, Brasil.

EVSUKOFF, A. G., ALMEIDA P. E. M., 2003, "Sistemas Fuzzy", In: Rezende, S. O. (coord), "*Sistemas Inteligentes – Fundamentos e Aplicações*", 1ª. Edição, capítulo 7, São Paulo, Brasil, Ed. Manole

FOGEL, G.B, CORNE, D. W., 2003 "An Introduction to Evolutionary Computation for Biologists" In: Fogel, G.B and Corne, D. W. (eds), "*Evolutionary Computation in Bioinformatics*", Nova Iórque, EUA, Morgan Kaufmann Publishers.

FREITAS, A.A., 2000, *Tutorial on Data Mining with Evolutionary Algorithms*, Genetic and Evolutionary Computation Conference (GECCO-2000). Las Vegas, EUA, Jul.

GESTEL, T. V., SUYKENS, J., BAESENS, B., *et al.*, 2001, "Benchmarking least squares support vector machine classifiers", *Machine Learning,* v. 54 , n. 1 (Jan), pp. 5-32, Kluwer Academic Publisher.

GIBSON, D. ; KLEINBERG, J., RAGHAVAN, P., 2000, "Clustering categorical data: an approach based on dynamical systems" , *The VLDB Journal* v. 8, n. 2 (Mar), pp 222–236, Springer Verlag Ed.

GILMOUR S. J.; ZARKA, D. G., 1998, "Low Temperature Regulation of the Arabidopsis CBF family of AP2 transcriptional activators as an early step in cold-induced COR gene expression", *The Plant Journal* v. 16 n. 4 (Apr) pp. 433-442, Blackwell Science Ltd.

GORDON, A. D., 1981 "*Classification*", 1ª Edição, Boston, EUA, Chapman and Hall Editors.

GRAY, S.; LEVINE, M., 1996, "Transcriptional Repression in Development", *Current Opinion in Cell Biology*, v. 8, n. 2, pp.358-364.

GRIFFITHS, A. J. F; MILLER, J. H.; SUZUKI, D. T. *et al.*, 2000, "An Introduction to Genetic Analysis" 7ª Edição, Nova York, EUA, Ed. W. H. Freeman.

GUHA, S., RASTOGI, R., SHIM, K., 2000, "Rock: A robust clustering algorithm for categorical attributes." *Information Systems*, v. 25, n. 5 (May), pp. 345-366.

GUPTA, M., YAMAKAWA, T., 1991, *Fuzzy Logic in Knowledge Based Systems, Decision and Control*, 1ª. Edição, Amsterdã, Holanda, Elsevier Science Publishers B. V.

HAN, J., KAMBER, M., 2001, *Data Mining: Concepts and Techniques*, 1 ed., California, EUA, Morgan Kaufmann.

HANISCH, D. ; ZIEN, A., 2002, "Co-clustering of Biological Networks and Gene Expression Data", *Bioinformatics* v. 18 Suppl. 1, pp. 145-154, Oxford University Press.

HAUTANIEMI, S., 2003, *Studies Of Microarray Data Analysis With Applications For Human Cancers*, DSc. Dissertation, Tampere University of Technology, Finlândia.

HERRERA, F., MAGDALENA, L, 1997, "Genetic Fuzzy Systems – a Tutorial", *Proceedings of the Seventh IFSA World Congress (IFSA97)*, pp. 93-121 Praga, República Tcheca, Jun.

HETLAND, M. L, SAETROM, P., 2002, *Temporal Rule Discovery using Genetic Programming and Specialized Hardware*", Technical Report IDI, Oslo, Noruega.

HOLLAND, J. H., 1975 "Adaptation in Natural and Artificial Systems", 1ª Edição, Cambridge, EUA, The MIT Press.

HWANG, D.; SCHMITT, W. A., 2002, "Determination of minimum sample size and discriminatory expression patterns in microarray data", *Bioinformatics*, v. 18, n. 9 (Sep.), pp 1184-1193, Oxford University Press.

JIANG, D., ZHANG, A., 2002, *Cluster Analysis for Gene Expression Data: A Survey*, Technical Report 2002-06, State University of New York at Buffalo, EUA.

KAUFFMAN, S. A., 1969, "Metabolic stability and epigenesis in randomly constructed genetic nets,", *J. Theoret. Biol.*, v. 22, n. 3 (Mar), pp. 437–467.

KAWAJI, H., YAMAGUCHI, Y., MATSUDA, H. *et al.*, 2001, "A Graph-Based Clustering Method for a Large Set of Sequences Using a Graph Partitioning Algorithm",*Genome Informatics*, v. 12, n. 1 (Jan), pp. 93–102, Universal Academy Press.

KEOGH, E., KASETTY, S., 2003, "On the Need for Time Series Data Mining Benchmarks: A Survey and Empirical Demonstration", *Data Mining and Knowledge Discovery*, vol. 7, n. 2 (Feb.) 349–371, Kluwer Academic Publishers.

KIM, Y.; STREET, W. N., MENCZER, F., 2002, "Evolutionary model selection in unsupervised learning", *Intelligent Data Analysis* vol. 1 n. 6 (Nov) pp.. 531–556, IOS Press.

KOZA, J. R., 1992, "Genetic Programming: On the Programming of Computers by Means of Natural Selection", 1ª. Edição, Cambridge, EUA, The MIT Press.

KOZA, J. R.; MYDLOWEC, W., LANZA, G., *et al*., 2001, "Reverse Engineering Of Metabolic Pathways From Observed Data Using Genetic Programming", In: *Proceedings of the Pacific Simposium on Bio-Informatics (PSB'01)*, pp. 434-445, Havaí, EUA, Jan.

LEE, B.; YEN, J., YAN, L. *et al*., 1999, "Incorporating qualitative knowledge in Enzyme kinetic models using fuzzy logic", *Biotechnol Bioeng.* v. 62, n. 6 (Mar.), pp. 722-729

LEE, H. K. H., 2000, *Model Selection for Neural Network Classification*, Technical Report 2000-1, Duke University, EUA.

LIANG, S., FUHRMAN, S., SOMOGYI, R., 1998, "REVEAL, a general reverse engineering algorithm for inference of genetic network architectures" In *Proceedings of the Pacific Symposium on Biocomputing (PSB'98)* pp. 18-29, Havaí, EUA, Feb.

LINDEN, RICARDO; BHAYA, AMIT, 2002, "Reverse Engineering of Genetic Networks under the Boolean Networks Model using Variable-Length Genetic Algorithms", *Proceedings of the XIII Artificial Neural Networks in Engineering*, pp. 535-541, St. Louis, EUA, Nov.

LINDEN, RICARDO; BHAYA, AMIT, 2002, "Reverse Engineering of Genetic Networks under the Boolean Networks Model using Variable-Length Genetic Algorithms", *Anais do I Workshop Brasileiro de Bioinformática*, pp. 94-96, Gramado, Brasil, Oct.

MANGASARIAN, O., LEE, Y., 1999, "*SSVM: A smooth support vector machine for classification*", Data Mining Institute Technical Report 99-03, EUA.

MCADAMS, H.H.; SHAPIRO, L., 1995, "Circuit simulation of genetic networks", *Science*, v. 269 n. 8 (Feb) pp 650-656.

MENDES, R., VOZNIKA, F. B., FREITAS, A. et al., 2001, "Discovering Fuzzy Classification Rules with Genetic Programming and Co-Evolution", In: *Proceedings of the 5th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'01)*, pp. 314-325, Freiburg, Alemanha, Sep.

MEYER, D.; LEISCH, F. E HORNIK, K., 2002, "*Benchmarking Support Vector Machines*", Technical Report nº 78, Report Series, Vienna University of Economics and Business Administration, Áustria

MICHALSKI, R. S., MOZETIC, I., HONG, J., et al., 1986, "The Multi-Purpose Incremental Learning System AQ15 and its Testing Application to Three Medical Domains" In *Proceedings of the Fifth National Conference on Artificial Intelligence*, pp. 1041-1045,Philadelphia, EUA, Aug.

MITCHELL, M., 1996 "*An Introduction to Genetic Algorithms"*, 1ª. Edição, Cambridge, Massachussets, The MIT Press.

MOORE, D. S., MCCABE, G.P, 2002, "*Introdução à Prática da Estatística*", Rio de Janeiro, Brasil, Editora LTC, 2002.

SUÁREZ, A., 2001, "Using All Data to Generate Decision Tree Ensembles", Tese de MSc., Universidade Autónoma de Madrid, Madri, Espanha

MURPHY, P. M., AHA, D. W., 1996, *UCI repository of machine learning databases [machine-readable data repository*]. Technical Report, University of California at Irvine, Department of Information and Computer Science, California, EUA.

NAUCK, D. AND KRUSE, R., 1995 "NEFCLASS – A Neuro-Fuzzy Approach for the Classification of Data" , *Proc. of the 1995 ACM Symposium on Applied Computing*, pp. 461-465, Nashville, EUA, Feb.

OHLER, U., 2002, *Computational Promoter Recognition in Eukaryotic Genomic DNA*, PhD Dissertation, Universidade Erlanger-Nürnberg, Alemanha

O'DEA, P.; GRIFFITH, J. E O' RIORDAN, C., 2001, "Combining Feature Selection and Neural Networks for Solving Classification Problems", In: *Proceedings of the 12th Irish Conference on Artificial Intelligence & Cognitive Science (Lecture Notes in Artificial Intelligence)*,pp. 157-166, Springer-Verlag Publisher Dublin, Irlanda, Sep.

ONODA, T., RATSCH, G., AND MULLER, K.R., 2001, "Soft margins for adaboost" *Machine Learning*, vol. 42 n. 3 (Mar), pp.287–320, Kluwer Academic Publisher.

PALMER, C. R., FALOUTSOS, C., 2003, "Electricity Based External Similarity of Categorical Atributes", In: *Proceedings of the Seventh Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 120-126, Seul, Coréia, Apr.

PEDRYCZ, W. E GOMIDE, F., 1998, "An Introduction to Fuzzy Sets – Analysis and Design", 1ª. Edição, Cambridge, EUA, MIT Press.

PE'ER, D.; REGEV , A. ; TANAY, A., 2002 "Minreg: Inferring an active regulator set", *Bioinformatics* v. 18, Suppl. 1 (Dec), pp. S258–S267

PIASECZNY, W., SUZUKI, H., SAWAI, H., 2004, *Chemical Genetic Programming – The Effect of Evolving Aminoacids*, Technical Report, ATR Human Information Labs, Tóquio, Japão

POZDNOUKOV, A.; BENGIO, S., 2004, *Invariances in Kernel Methods: from samples to objects*, Relatório Técnico, Instituto Pascal, Berna, Suíca.

PIETROKOVSKI, S., HENIKOFF, S., 1997, "A helix-turn-helix DNA-binding motif predicted for transposases of DNA transposons", *Molecular Genetics* v. 254 n. 3 (Mar.), pp. 689-695, Springer Verlag Ed.

QUINLAN, J. R., 1986, "Induction of Decision Trees", *Machine Learning* v. 1, n. 1 (Jan) pp. 81-106, Kluwer Academic Publishers.

RIPLEY, B. D., 1993, "Statistical Aspects of Neural Networks" in Barndorff-Nielsen, O. E. ; Jensen, J. L. e Kendall, W. S. (eds), *Networks and Chaos – Statistical and Probabilistc Aspects*, 1a. Edição, cap. 4, Chapman & Hall Ed., Inglaterra,

RUGGIERO, M. A. G.; LOPES, V. L. R., 1996, "*Cálculo Numérico – Aspectos Teóricos e Computacionais*", 2ª. Edição, Rio de Janeiro, Brasil, Ed. Makron Books.

SAKAMOTO, E., IBA, H., 2001, “Inferring a System of Differential Equations for a Gene Regulatory Network by using Genetic Programming”, In: *Proceedings of the 2001 Congress on Evolutionary Computation (CEC’01)*, pp. 720-726, Seul, Coréia do Sul, Maio.

SOCIEDADE BRASILEIRA DE DIABETES, 2004, “Diabetes e o Peso”, disponível no site http://www.diabetes.org.br/diabetes/atividade/peso.php, Brasil.

SCHEAFFER, R., 1999, *Categorical Data Analysis*, Technical Report - NCSSM Statistics Leadership Institute, Carolina do Norte, EUA

SCHRAGER, J; LANGLEY, P., POHORILLER, 2002, “Guiding revision of regulatory models with expression data”, In: *Proceedings of the Pacific Symposium on Biocomputing (PSB’02)*, pp. 486-497, Havaí, EUA, Feb.

SHEDDEN, K. AND COOPER, S., 2002, “Analysis of cell-cycle-specific gene expression in human cells as determined by microarrays and double-thymidine block synchronization”, *Proceeding of the National Academy of Sciences (PNAS)*, vol. 99 n. 7 (Feb.), pp. 4379-4384.

SINGAL, P. K, MITRA, S., PAL, S. K., 2001, “Incorporation of Fuzziness in ID3 and Generation of Network Architecture”, *Neural Comput & Applic* Vol. 10 n. 1 (Jan), pp. 155–164, Springer-Verlag London Limited.

SMOLEN, P, BAXTER, D, BYRNE, J. D., 2000, “Modeling Transcriptional Control in Gene Networks—Methods, Recent Results, and Future Directions”, *Bulletin of Mathematical Biology* v. 62, n. 2 (Mar), pp. 247–292.

SONTAG, E. ; KIYATKIN, A. E KHOLODENKO, B. N., 2004, “Inferring Dynamic Architecture of Cellular Networks Using Time Series of Gene Expression, Protein and Metabolite Data”, *Bioinformatics*, no prelo, Oxford University Press.

VAN SOMEREN, E.P.; WESSELS, L.F.A ; REINDERS, M.J.T., 2000, "Linear Modeling of Genetic Networks from Experimental Data", In: *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology* , pp. 355-366, La Jolla, EUA, Aug.

SPIETH, C.; STREICHERT, F., 2004, *Optimizing Topology and Parameters of Gene Regulatory Network Models from Time-Series Experiments*, Technical Report, Universidade de Tübingen, Alemanha.

STOCKBURGER, D., 1996, *Introductory Statistics*, Online textbook, available at http://www.psychstat.smsu.edu/sbk00.htm

SUER, G. A., BADURDEEN, F., THANGALEVU, B., 2002, "Capacitated lot sizing by using multi-chromosome crossover strategy", in *Proceedings of the XII Conference on Artificial Neural Networks in Engineering*, pp. 281-286, St. Louis, EUA, Nov.

SZALLASI, Z., LIANG, S, 1998,: "Modeling the normal and neoplastic cell cycle with realistic boolean genetic networks: their application for understanding carcinogenesis and assessing therapeutic strategies", In: *Proceedings of the Pacific Symposium on Biocomputing (PSB '98)*, pp 66-76. Havaí, EUA, Feb.

TAN, M., ESHELMAN, L., 1988, "Using weighted networks to represent classification knowledge in noisy domains", In: *Proceedings of the Fifth International Conference on Machine Learning*, pp. 121-134, Ann Arbor, MI, Aug.

THEIL, H., 1971, "Applied Economic Forecasting", 1ª. Edição, Amsterdã, Holanda, North Holland Publishing Co.

THOMAS, R., 1998, "Laws for the dynamics of regulatory networks", *International Journal Developmental Biology* v. 42 n. 5 (May.) pp. 479-485.

TOIVONEN, H., 1996, "Sampling Large Databases for Association Rules" In: *Proceedings of 22th International Conference on Very Large Data Bases*, pp. 134-145, Morgan Kauffman Publishers, Bombaim, Índia, Sep.

WAGNER, A., 2001, "How to reconstruct a large genetic network from n gene perturbations in fewer than n^2 easy steps", *Bioinformatics*, v. 17, n. 12 (Dec), pp 1183-1197.

WANG, L., 1994, "Adaptative Fuzzy Systems and Control – Design and Stability Analysis", 1 Edição, Nova Iórque, EUA, Ed. Prentice-Hall

VENET , D.; MAENHAUT, C., BERSINI, H., 2001 – "Modeling and determination of regulation of gene expression: the binary switch model", In: *Proceedings of the 2nd. International Conference on Systems Biology*, pp. 239-247, Pasadena, EUA, Nov.

YANG, Z. R.; THOMSON, R., 2003, "Searching for discrimation rules in protease proteolytic cleavage activity using genetic programming with a min-max scoring function", In: *BioSystems* v. 72 n. 2 (Feb.) pp.159-176, Elsevier Ireland Ltd., Irlanda

ZHANG, Y., FU A. W., CAI, C. H. et al., 2000, "Clustering Categorical Data", In: *Proceedings of the 16th International Conference on Data Engineering*, pp. 305-316 San Diego, EUA, Feb.

ZHOU, C., XHIAO, W., TIRPAK T. M., *et al.*, 2002 "Discovery of Classification Rules by Using Gene Expression Programming" In *Proceedings of the International Conference on Artificial Intelligence*, pp. 1355-1361, Chicago, Jun.

ZIEN, A., FLUCK, J., ZIMMER, R. et al., 2002, "Microarrays: How Many Do You Need?", In *Proceedings of RECOMB02*, pp. 321-330, Seattle, EUA, Jan.

Apêndice A– Aplicações em Classificação

A.1 Introdução

O aprendizado indutivo consiste em estabelecer hipóteses sobre exemplos fornecidos por um processo externo ao sistema de aprendizado e pode ser dividido em dois tipos:

- *Supervisionado*: neste caso, é fornecido ao algoritmo de aprendizado um conjunto de exemplos de treinamento para os quais o resultado esperado (rótulo de classe, no caso da classificação) é conhecido.
- *Não supervisionado*: nenhuma estrutura é imposta a priori aos dados, cabendo ao algoritmo classificador descobrir uma hipótese subjacente aos dados que permita classificá-los de alguma forma lógica.

No primeiro caso, o problema a ser resolvido é conhecido como algoritmo de classificação para rótulos de classe discretos e de regressão, para valores contínuos. Já no segundo caso, o problema a ser resolvido é denominado agrupamento ou clustering (GORDON, 1981) e (BARANAUSKAS *et al.*, 2003).

Neste capítulo será explicado como se aplica o algoritmo de aprendizado aqui proposto para o problema de classificação, discutindo alguns exemplos de aplicação em conjuntos que são normalmente usada na literatura da área.

Segundo HAN & KAMBER (2001), os algoritmos de classificação podem ser avaliados e comparados utilizando-se os seguintes critérios:

- **precisão**: é a capacidade de classificar corretamente novos elementos, usando o modelo definido a partir dos dados de treinamento. O modelo proposto será avaliado em termos deste critério e também usando-se modelos naïve como base de comparação, como será visto neste apêndice;
- **rapidez**: refere-se tempo gasto na fase de aprendizado e na utilização do modelo. É importante entender que algoritmos evolucionários são técnicas de busca que mantêm grandes populações que têm que se avaliadas contra toda a base de treinamento, sendo por isto inerentemente mais lentas que outros métodos como as máquinas de vetor de suporte. Logo, a rapidez não pode ser um critério fundamental quando se pretende utilizar um algoritmo evolucionário;
- **escalabilidade**: é a capacidade de construir classificadores na presença de grandes bases de dados. Este conceito não é conceitualmente aplicável ao algoritmo desta tese, posto que está se propondo uma técnica a ser aplicada para bases de dados escassas. Para bases de grande porte, sugere-se aplicar qualquer técnica estatística estabelecida e largamente utilizada (MOORE, 2002)
- **robustez**: é a capacidade de realizar decisões corretas na presença de dados com ruído ou incompletos;
- **interpretabilidade**: refere-se ao nível de entendimento do modelo construído. No caso do modelo proposto, o uso de regras fuzzy permite que se crie uma estrutura pseudo-linguística que tem grande similaridade com a linguagem utilizada de forma cotidiana na comunicação de idéias pelas pessoas. Assim, as regras obtidas possuem grande interpretabilidade para o usuário final.

A.2 Mudanças na função de avaliação

A função de avaliação utilizada para classificação não pode ser a mesma usada para a previsão, pois esta se concentra em erros percentuais enquanto que no caso da classificação, é necessário apenas optar entre dois ou mais valores.

As classes previstas nesta tese são determinadas por variáveis Booleanas. Isto é, se temos duas classes apenas, usar-se-á uma variável Booleana para discriminá-las. Se temos mais de duas classes, serão usadas um número de variáveis Booleanas igual ao número de classes para separá-las.

Tendo em vista que um sistema fuzzy é usado para prever as classes e que opta-se por um defuzzificador do tipo médio dos máximos (MoM), as variáveis Booleanas usadas deixam de ser consideradas como assumindo valores 0 ou 1 para assumir os valores 1 ou 2. Isto mantém seu caráter essencialmente binário mas permite que o limite inferior passe a contribuir para a média de uma forma efetiva, enquanto que se usássemos o zero como um dos limites, sua contribuição para a média seria sempre zero. Isto decorre do fato de que o algoritmo MoM utiliza a seguinte fórmula para realizar a defuzzyficação:

$$y = \sum_{p} \mu_p \max_p$$

onde:

- y é a saída gerada
- p é o número de conjuntos fuzzy adotados para a divisão do universo de discurso da variável de saída
- μ_p é a pertinência calculada da variável de saída ao conjunto p pela aplicação da base de regras
- $\max_p$ é o valor máximo do conjunto fuzzy p.

Assim, se o valor máximo for igual a zero, a parcela de contribuição deste conjunto, não importando o valor calculado para μ_p, será sempre zero. Isto não é relevante em uma situação em que usamos apenas dois conjuntos que possuem pertinência complementar

($\mu_1 = 1 - \mu_0$), pois neste caso a conta obteria exatamente o mesmo resultado. Em qualquer outra situação, seja com número maior de conjuntos ou com formatos distintos da função de pertinência, a presença do máximo em zero causará problemas no cálculo do valor defuzzyficado.

No caso de haver apenas duas classes, a defuzzificação da variável de categoria gerará um número entre 1 e 2. Para este número, cria-se então uma zona de classificação errônea, que consiste em [1,5-ε ; 1,5+ε], onde ε é um parâmetro fornecido ao algoritmo. Todos os valores defuzzificados que resultarem em um valor neste intervalo serão tratados como uma falha de classificação, isto é, uma incapacidade do algoritmo de classificar corretamente a tupla.

Os valores obtidos maiores que 1,5+ε serão tratados como pertinentes à classe 2, enquanto que os valores obtidos que sejam menores que 1,5-ε serão tratados como pertinentes à classe 1.

Outra abordagem possível seria não defuzzyficar as pertinências, simplesmente atribuindo a tupla à classe que obtiver a maior pertinência. As duas abordagens são bastante similares, dado que foram utilizados conjuntos criado a partir de uma divisão uniforme do universo de discurso. Isto quer dizer que a partir do ponto 1,5, a pertinência da classe 2 é maior que a da classe 1, e antes disto o inverso ocorre, e a classe 1 predomina. Em outras situações, com mais de duas classes ou com uma divisão não uniforme do espaço de estados, como a baseada nos dados descrita em (SINGAL *et al.*, 2001), as duas abordagens poderiam ser comparadas.

A função de avaliação do indivíduo consiste no número de acertos de classificação realizados.

É hábitual na comunidade de pesquisadores de classificação usar o número de erros multiplicado por um valor negativo como forma de punição dentro da função de avaliação. Assim, sua função de avaliação torna-se:

$F = c_1 * n_{acertos} - c_2 * n_{erros}$, onde c_1 e c_2 são constantes positivas a determinar.

Entretanto, este tipo de acréscimo à função de avaliação não acrescenta nenhum tipo de informação a esta, pois tendo em vista que n_{erros} é igual a $n\text{-}n_{acertos}$ (onde n é o número de exemplos dentro do conjunto de treinamento), obtemos o seguinte:

$$F = c_1 * n_{acertos} - c_2 * (n - n_{acertos})$$

$$F = (c_1 + c_2) * n_{acertos} - c_2 * n$$

$$F = k_1 * n_{acertos} - k_2 * n$$

O que implica que este tipo de acréscimo consiste apenas em multiplicar a função originalmente usada e somar uma constante negativa a esta. Além de não fornecer nenhum tipo de informação adicional, este tipo de acréscimo ainda cria uma dificuldade adicional de fornecer valores negativos quando

$$n_{acertos} < \frac{k_2 * n}{k_1}$$

Um ponto fundamental relativo à função de avaliação usada em um algoritmo genético é que nenhum elemento deve ter avaliação negativa ou zero. Isto faria com que a soma das avaliações diminuisse, alterando a roleta e fazendo com que houvesse mais de um elemento possivelmente selecionado dada a escolha do mesmo valor de intervalo. Isto quer dizer que, se um elemento *k* possui avaliação negativa, tem-se a seguinte condição para uma população de *n* elementos mantida por uma algoritmo genético de função de avaliação *f*:

$$\sum_{i=1}^{n} f(i) < \sum_{i=1;i\neq k}^{n} f(i)$$

Simplificando, pode-se afirmar que adicionar uma avaliação negativa diminui o total da função de avaliação. Isto resulta em dois elementos poderem ser escolhidos através do mesmo resultado para um sorteio da roleta e, sob certas circunstâncias, no fato de que o segundo destes elementos nunca será efetivamente escolhido.

Esta questão pode ser facilmente entendido com o seguinte exemplo. Imagine que temos 100 exemplos, com $k_1 = k_2 = 1$, e três cromossomos. O primeiro classifica corretamente 60 elementos, o segundo 30 elementos e terceiro, 70 elementos. Isto resulta que a fitness do primeiro será igual a 20, a do segundo a –40 e a do terceiro, a 40.

O somatório das funções de avaliação será igual a 20 e por conseguinte a roleta selecionará um número de 1 a 20, selecionando eternamente o primeiro cromossomo, acabando com a variabilidade genética da população de cromossomos.

Assim, torna-se imperioso eliminar a constante acrescida à função de avaliação o que nos leva à função originalmente usada, multiplicada por uma constante positiva. Tendo

em vista que o acréscimo de uma penalização resulta apenas em um deslocamento vertical sem aumento do poder discriminatório da função de avaliação, usou-se apenas a informação quanto ao número de acertos na função de avaliação de nosso programa genético.

No caso de termos mais de duas categorias, poderia ser usada uma variável contínua para descrever as categorias às quais um elemento pertence. Por exemplo, se a variável assumisse o valor *i* (i=1,..., n_c, , onde n_c é o número de categorias existentes), o dado pertenceria à categoria *i*. Entretanto, isto criaria uma situação em que uma categoria seria mais próxima de outra, sem que isto tenha necessariamente um reflexo na estrutura dos dados. Por exemplo, seria criada, de forma artificial, uma proximidade maior entre as categorias 1 e 2 do que entre as categorias 1 e 3.

Para evitar a introdução deste tipo de estrutura nos dados, pode-se optar pela utilização de variáveis Booleanas. Neste caso, será necessário mais de uma variável Booleana para descrever todas as categorias. As categorias poderiam ser descritas usando-se $\lfloor \log_2 n_c \rfloor$ variáveis, mas isto também criaria uma proximidade artificial entre elas. Por exemplo, a categoria três (representada pelo número Booleano 011) seria mais próxima da categoria 1 (representada pelo número Booleano 001) do que da categoria 4 (representada pelo número Booleano 100) já que a diferença entre 1 e 3 é apenas um bit enquanto que entre 1 e 4 é de 3 bits.

Assim, optou-se por uma estrutura em que foram usadas exatamente n_c variáveis Booleanas. Para designar a categoria i, apenas a variável i assume valor 2, enquanto que todas as outras variáveis assumem valor 1. Isto pode ser um problema se lidarmos com variáveis com números grandes de categorias, mas os problemas de classificação de ordem prática tendem a lidar com pequenos números de categorias, da ordem de 2 a 5, o que faz com que esta abordagem não cause aumento expressivo no número de variáveis em avaliação.

Escolhido o número de variáveis, é necessário definir como o resultado da defuzzyficação será tratado em termos de categorias. Da mesma forma que no caso de duas variáveis, o processo de defuzzyficação faz com que cada uma das variáveis assuma um valor contínuo no intervalor [1,2], mas é desejável tratar apenas uma das variáveis como se

tivesse valor dois (categoria efetiva do dado) e todas as outras como se tivessem valor 1 (categorias às quais o dado não pertence).

Para fazer isto, usamos o conceito de dominância. Consideramos que a variável que possui o maior valor é a categoria corrente do dado sendo classificado, desde que a diferença deste valor para todos os outros seja maior do que um parâmetro ε definido pelo usuário. Podemos formular tal abordagem matematicamente dizendo que o dado pertence à categoria i se e somente se $\forall j = 1,2.,...,n_c, j \neq i \Rightarrow v_i - v_j \geq \varepsilon$, onde v_i e v_j significam respectivamente os valores assumidos pelas variáveis i e j. Se isto não for verdade para nenhuma variável i, então o dado é considerado como sendo uma classificação errônea.

Uma vez obtidos os valores para todas as variáveis, usamos o mesmo critério que usamos para o caso de termos apenas uma variável, isto é, o número de dados classificados corretamente é o valor da função de avaliação para o conjunto de regras em questão.

Existem artigos tais como (EGGERMONT *et al.*, 2004a) que utilizam um critério fuzzy para determinar a avaliação de uma classificação, usando um critério que matematicamente pode ser explicitado da seguinte forma:

$$fitness_{classificação}(x) = \sum_{r \in conj.treinamento} (1 - \rho(r))$$

Esta função de avaliação usa a pertinência designada para cada um dos conjuntos de forma que se o dado *x* pertence à classe *y*, mas esta função de classificação associa qualquer pertinência ρ à classe y menor do que 1, um erro de classificação de 1-ρ. Da mesma forma, qualquer pertinência diferente de 0 em qualquer outro conjunto fuzzy que não y é tratado como erro e integra o somatório da função de avaliação.

Tendo em vista que, tanto no caso desta tese como do artigo em questão, se está lidando com dados que pertencem a uma única classe, sem nenhum tipo de gradação, este tipo de função acrescenta muito pouca informação em relação a um critério claro de dominância. No fundo, o que esta métrica procura é forçar o algoritmo evolucionário usado a buscar regras que impliquem em uma dominância maior (de preferência, de tal forma que o resultado das regras fuzzy retorne uma classificação praticamente crisp).

A.3 Avaliação do desempenho de um algoritmo de classificação

Nos últimos anos, vários novos algoritmos de classificação foram apresentados em conferências e revistas. Tornou-se um aparente consenso que para avaliar a capacidade destes algoritmos, estes deveriam ser testados com alguns conjuntos de dados disponíveis no site da UC-Irvine (MURPHY *et al.*, 1996).

Assim, estes conjuntos de dados se tornaram uma espécie de "benchmark" na literatura, e se tornou um paradigma *de facto* que qualquer novo método deve mostrar que é superior a trabalhos anteriores nos quesitos velocidade e/ou precisão quando trabalhando nestes conjuntos de dados.

Uma característica comum à maioria dos artigos que seguem este consenso é que eles omitem certas informação inerentes aos conjuntos analisados que colocariam os resultados em um contexto, deixando claro a validade das taxas de acerto, ao invés de mostrá-las como números puros.

Seria necessário analisar os resultados não só em termos das percentagens de classificação correta, mas também em termos das características inerentes aos conjuntos de dados. Por exemplo, certos conjuntos de dados, como o da Íris, são extremamentes simples para classificação e resultados de 95% podem ser considerados ruins, enquanto que em outros conjuntos, tais como o da Diabetes, este nível de resultado parece ser inatingível.

Assim, para dar maior perspectiva para os resultados obtidos aqui, fez-se uma comparação dos resultados obtidos com classificadores naïve e árvores de decisão treinadas com o algoritmo ID3 (QUINLAN, 1986).

O algoritmo ID3 tradicional pode ser descrito como uma heurística gulosa baseada na entropia condicional ($H(Y \mid X)$) e é dada pelo seguinte pseudo-código:

1. *Calcule a entropia condicional da categoria em relação a cada um dos atributos, de acordo com a seguinte fórmula:*

$$H(Y \mid X) = \sum_j \Pr(X = \upsilon_j) H(Y \mid X = \upsilon_j), \; e$$

$$H(Y \mid X = \upsilon_j) = -\sum_k \Pr(Y = \upsilon_k) * \log_2(\Pr(Y = \upsilon_k))$$

2. *Selecione o atributo com a menor entropia*
3. *Crie os subconjuntos para cada valor do atributo*
4. *Para cada subconjunto:*
    - *Se todos os elementos do subconjunto pertencem à mesma classe, crie uma folha e encerre para aquele subconjunto.*

   Senão, volte para o passo 1.

Ambos os previsores simples são considerados nesta tese como os "padrões-ouro" que devem ser superados por quaisquer algoritmos de classificação supostamente melhores.

Para realizar esta comparação, introduzimos uma versão modificada da estatística U de Theil (doravante denominada U-Theil), que é tradicionalmente usada na área de previsão de séries temporais.

Existem duas versões do U-Theil para resumir a precisão de um algoritmo de previsão. Se um determinado método prevê um conjunto P_i de mudanças na série, estas podem ser comparadas com as mudanças efetivas na série (A_i). O coeficiente U-Theil pode ser dado então por:

$$U = \sqrt{\frac{(P_i - A_i)^2}{A_i^{\,2}}} \qquad (1)$$

Se as previsões são perfeitas, o valor de U é zero. O modelo naïve que prevê que não haverá nenhuma alteração obtém um valor de U igual a 1. Assim, modelos que tenham uma performance melhor do que o método naïve devem obter valores de U no intervalo [0,1[(THEIL, 1971).

Uma das grandes vantagens deste coeficiente é que ele é independente dos valores reais presentes na série, além de ser definido como um número sem unidades.

O coeficiente modificado proposto aqui tem as mesmas características e o mesmo espírito, isto é, a comparação do modelo previsor proposto com um modelo naïve. No caso de algoritmos de classificação, pode-se comparar a taxa de erro obtida no modelo proposto (E_p) com a taxa de erro obtida no modelo naïve (E_n). O coeficiente U-Theil modificado seria dado então por:

$$U = \frac{E_p}{E_n} \tag{2}$$

Não é necessário tirar o módulo das medidas pois tendo em vista que elas consistem em uma taxa de erro, naturalmente elas estão contidas no intervalo [0,1].

Assim como no caso das séries temporais, este coeficiente retorna um valor no intervalo [0,1[para modelos que tenham desempenho superior ao modelo naïve, sendo que o valor obtido também é sem unidades e independentes dos valores reais existentes no conjunto de dados em estudo.

Para cada aplicação existente, o modelo naïve é distinto. É necessário estudar as características implícitas nos dados e o corpo de conhecimento da área que gerou o conjunto, e não simplesmente se aplicar um algoritmo de classificação às cegas e esperar por um bom resultado.

Na maioria dos casos estudados para esta tese, o modelo naïve adotado consistiu em uma reordenação dos dados seguida de uma inspeção visual, de forma a atribuir de forma imediata uma classe a cada tupla de dados. Este é um modelo subjetivo e geralmente pouco eficiente. A sua baixa eficiência o torna um bom modelo de comparação, posto que todos aqueles que não o superarem devem avaliar de forma profunda a razão desta falha. Os casos em que o modelo usado como base de comparação é distinto são claramente indicados no texto da seção apropriada.

A utilização desta métrica serve para contextualizar os resultados obtidos. Como dito acima, muitos trabalhos analisam o desempenho de seus algoritmos através da simples comparação com os resultados obtidos. Isto pode levar o leitor/usuário de tais trabalhos a considerar como positivo e valioso um resultado de pouca conseqüência real.

É extremamente importante que a comparação com algoritmos simples seja efetuada e que seja feito um estudo da área científica que gerou os dados sob análise. Isto permite

que sejam evitados estudos desnecessários, onde o ser humano já é um classificador mais preciso ou onde o conhecimento disponível demonstra que precisão absoluta é teoricamente inatingível.

Estes conceitos serão aplicados em todos os testes aplicados nos conjuntos de dados usados como “benchmarks” do algoritmo proposto nesta tese.

A.4 Aplicações numéricas do algoritmo proposto

Podemos agora tentar aplicar nosso algoritmo a alguns conjuntos de dados tradicionalmente usados em problemas de classificação. Ambos os conjuntos obtidos nesta seção foram retirados do conhecidos repositório de UC-Irvine (endereço da internet *http://www.sgi.com/tech/mlc/db*).

A.4.1 Íris

Este pequeno conjunto de dados, consistindo de 150 exemplos, é muito conhecido e utilizado em vários trabalhos como uma espécie de *benchmark* para modelos de classificações. Os dados consistem em 4 colunas de dados dados por:

- Comprimento da sépala, em cm.
- Largura da sépala, em cm.
- Comprimento da pétala, em cm
- Largura da pétala, em cm

Os dados são divididos em 3 classes diferentes: Iris Setosa, Iris Versicolor e Iris Virginica, dependendo da espécie da flor na qual foram medidos. As espécies são praticamente separáveis usando apenas os atributos de tamanho da pétala, como podemos ver na figura 6-1.

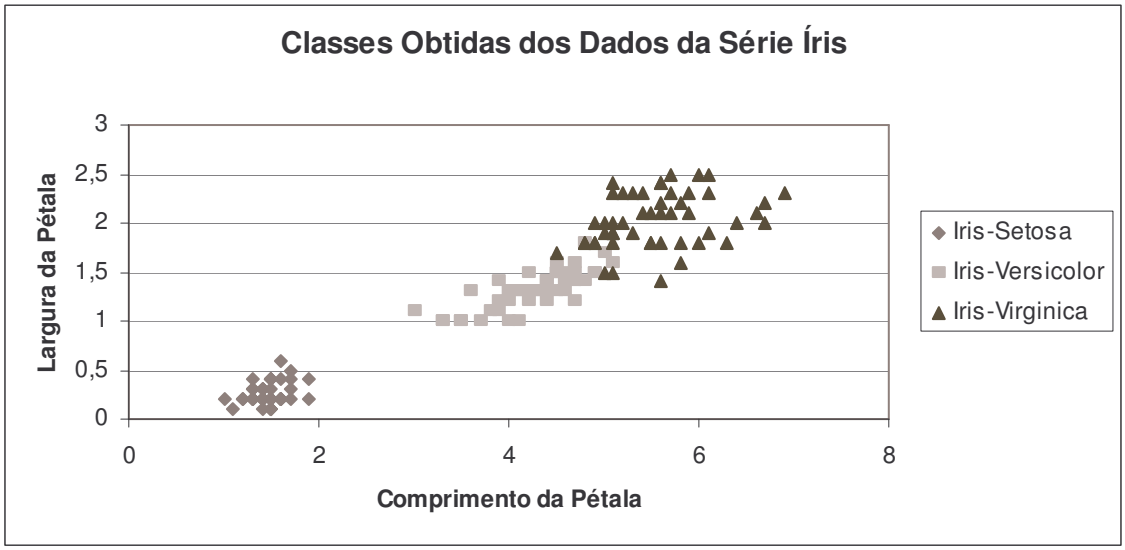


Figura 6-1 → Separação das classes com base nos dados de tamanho da pétala.

Os dados foram separados em dois conjuntos: o de treinamento com 80% dos dados (120 instâncias, consistindo de 40 instâncias de cada classe) e o de teste, com 20% dos dados (30 instâncias, 10 de cada classe).

Foram executadas 60 gerações de 100 cromossomos em cada população, usando uma probabilidade linearmente decrescente para o crossover (com valor inicial de 95%) e probabilidade de mutação de 5%. O módulo de população usou elitismo, mantendo vivas as duas melhores soluções de cada geração.

O módulo fuzzy foi configurado usando-se 5 conjuntos para cada variável numérica, criados a partir de uma divisão uniforme do universo de discurso, que, por sua vez, é obtido a partir dos dados reais. Foram definidos dois conjuntos fuzzy para cada uma das classes (denominados respectivamente *pertence* e *não pertence*), que foram definidas como três variáveis booleanas. Os operadores fuzzy utilizados foram do tipo min-max e uma média de uma regra por conjunto fuzzy.

Não foram tentados outros valores para nenhum dos parâmetros definidos tendo em vista o sucesso imediato atingido. Com apenas duas execuções do algoritmo foi atingida uma taxa de sucesso de 100% para a classificação do conjunto de testes. O conjunto de regras conseguido foi o descrito na tabela 6-1 a seguir.

a)	Pertence(Iris Virginica): Muito_Alto(Comprimento_Petala) AND Muito_Alto(Largura_Petala)
b)	Pertence(Iris_Versicolor): Médio(Comprimento_Petala) AND Muito_Alto(Largura_Petala) AND NOT [Médio(Largura_Sépala) AND NOT (Baixo(Largura_Sépala))]
c)	Pertence(Iris Setosa): Muito_Baixo(Comprimento_Pétala) AND NOT Muito_Alto(Largura_Pétala)

Tabela 6-1: Regras para classificação das flores Íris, conforme obtido pelo algoritmo proposto nesta tese.

A regra (b) apresenta uma relação importante entre a classe Iris Versicolor e a largura da sépala das flores. Se forem usadas apenas as duas variáveis de tamanho da pétala, existe uma área não classificável entre os dois conjuntos, como pode ser visto na figura 6-1. É necessário então acrescentar uma variável extra que as diferencie, o que foi automaticamente descoberto através da aplicação do algoritmo desta tese.

Para uma melhor verificação deste conjunto, foi feita uma rotação do conjunto de validação, isto é, o conjunto de dados foi separado em 5 conjuntos de tamanho igual (30 exemplos) e foram feitas mais quatro execuções do algoritmo, usando cada um dos conjuntos previamente não usados como conjuntos de testes. Este procedimento é idêntico ao processo de treinamento *five-fold*.

As regras obtidas em todas as execuções guardaram uma grande semelhança com as regras descobertas para o primeiro conjunto de treinamento, baseando-se nos atributos de tamanho da pétala. Isto já é previsto, tendo em vista que estes dois atributos guardam uma maior correlação com a separação de classes, conforme informado no banco de dados da UCI.

Para cada uma das execuções, usaram-se os mesmos parâmetros previamente definidos. Em um dos casos houve um erro de classificação (96,6% de acerto) , em outro ocorreram dois erros de classificação (93,3% de acerto), enquanto que nos outros dois casos foram necessárias apenas duas rodadas para que se obtivesse 100% de precisão no conjunto de validação.

No total, o algoritmo proposto nesta tese cometeu 3 erros de classificação em 150 exemplares (98% de acerto). Este valor compara positivamente com vários outros trabalhos da literatura, como por exemplo, o sistema neuro-fuzzy descrito em (NAUCK *et al.*, 1995), que conseguiu 95,7% de precisão na classificação.

Os três exemplos nos quais as regras determinadas pelo algoritmo aqui proposto incorreu em erro já foram discutidas em (RIPLEY, 1993) e são possivelmente erros originais de classificação. A base de dados das írises consiste de dados que foram originalmente coletados por Anderson em 1935, sendo que as plantas foram classificadas manualmente. Por conseguinte, é possível que haja erros na separação de classes originalmente determinada, ou que tenhamos indivíduos que são aberrantes para suas espécies. Estas aberrações podem ser devidas a fatores externos, tais como excesso ou falta de luz do sol, condições do solo, predadores ou qualquer outro tipo de interação da planta com o meio ambiente.

Em aplicações de classificação cujos dados são obtidos a partir da natureza, sempre existirão elementos cujas características são bastante diferenciadas em relação aos seus companheiros de conjunto. Estes elementos, denominados *outliers*, fazem com que a expectativa de se obter 100% de precisão neste tipo de aplicação seja normalmente irreal.

Este conjunto é extremamente simples e facilmente separável. Entretanto, dada a sua ampla difusão e uso, a correta descoberta de suas classes através da aplicação do algoritmo proposto é alentadora, justificando a aplicação deste algoritmo em exemplos mais complexos, descritos a seguir.

A.4.2 Diabetes

Estes dados foram originalmente compilados em 1988 pelo National Institute of Diabetes and Digestive and Kidney Diseases, e consiste em uma tabulação de vários fatores de saúde dos índios da tribo Pima, uma tribo que vive perto da cidade de Phoenix, no estado americano do Arizona. Por isto, este conjunto de dados também é conhecido também pelo nome de Pima Indians.

Este conjunto de dados consiste de 768 exemplos, divididos em um conjunto de treinamento (512 exemplos) e um conjunto de avaliação (256 exemplos). No conjunto de avaliação 155 dos indivíduos considerados (60,6% do total) não sofrem de diabetes.

O conjunto de dados consiste nas seguintes variáveis:

- Número de vezes que a pessoas ficou grávida.
- Concentração de glicose plasmática duas horas depois da alimentação em um teste de sobrecarga oral de glicose
- Pressão sangüínea diastólica (mm Hg)
- Espessura da dobra da pele do triceps (mm)
- Insulina em um teste de duas horas (mu U/ml)
- Índice de massa corporal (peso em quilos/(altura em metros)^2)
- Função de pedigree da diabetes
- Idade (anos)

Este conjunto é um exemplo típico de base de dados desbalanceada. Uma base de dados é desbalanceada quandouma classe é representada por um grande númer de exemplos, enquanto outras são representadas por apenas uns poucos (BATISTA *et al.*, 2004). Os classificadores tendem a dar grande ênfase aos acertos na classe dominante, quando a detecção de elementos pertencentes à classe presente em menor número pode ser fundamental, como no caso desta aplicação, onde o correto diagnóstico da doença é de fundamental importância.

A estimativa de desempenho pela acurácia assume que todos os tipos de erros ocorridos na classificação são iguais, assim como todos os benefícios dos acertos. Nos

casos de desbalanceamento, a acurácia tende a valorizar os classificadores com fraco desempenho em classes raras e, possivelmente, terá alta taxa de erro ao classificar novos elementos destas classes. (ESPINDOLA, 2004). Isto sugere a utilização de outros tipos de métricas, como por exemplo aquelas baseadas na matriz de confusão (tabela de contingência), que separa os resultados obtidos por um algoritmo de acordo com a correta classificação em cada uma das classes, identificadas como "positiva" e "negativa". A partir destas classificações, pode-se identificar quatro valroes em termos das classes previstas e das classes às quais cada tupla realmente pertence:

- VP: é a quantidade de elementos da classe positiva que foram previstos como pertencente a esta classe; os elementos são chamados de **verdadeiros positivos;**
- FP: é a quantidade de elementos da classe negativa que foram classificados como pretencentes à classe positiva; os elementos são chamados de **falsos positivos;**
- FN: é a quantidade de elementos da classe positiva que que foram classificados como pretencentes à classe ngativa; os elementos são chamados de **falsos negativos;**
- VN: é a quantidade de elementos da classe negativa que foram classificados como pretencentes à classe negativa; os elementos são chamados de **verdadeiros negativos;**

Na matriz de confusão, os valores da diagonal principal representam os acertos e os demais valores os erros. As medidas de desempenho mais comuns são (ESPINDOLA, 2004):

1. **sensitividade**: também chamada de *hit rate* e *recall*, ela avalia o quanto um classificador pode reconhecer os exemplos positivos e é definida por:

$$Sens = \frac{VP}{VP + FN}$$

2. **especificidade**: avalia o quanto um classificador pode reconhecer os exemplos negativos e é definida por:

$$Espec = \frac{VN}{FP + VN}$$

3. **precisão**: é a proporção de elementos classificados como positivos que de fato o são:

$$\Pr ec = \frac{VP}{FP + VP}$$

A variável de diagnóstico é binária consistindo em uma determinação do fato do paciente exibir sinais de diabetes de acordo com critério definidos pela Organização Mundial de Saúde (OMS). Este critério consiste no fato da glicose no sangue duas horas após a ingestão de alimentos ser de valor igual ou superior a 200 mg/dl em qualquer exame.

Existem vários previsores naïve possíveis, todos baseados em conhecimento médico básico. Um previsor é baseado no conceito de que mulheres com uma alta taxa de glicose no sangue (acima de 115) têm diabetes. Para o caso do conjunto de testes, este previsor naïve está correto em 100 dos 123 (81,3%) casos de mulheres que não têm diabetes em em 87 dos 133 (65,4%) casos em que as pacientes têm diabetes. A taxa geral de acerto é de 187 em 256 (73%).

Existem outros previsores naïve igualmente capazes de prever a diabetes. Estes são baseados nas variáveis de espessura da pele do tríceps e índice de massa corporal, que são variáveis que têm alto grau de correlação com a alta taxa de açúcar no sangue.

Outro previsor simples que pode ser considerado é uma implementação do algoritmo ID3. Tendo em vista que as variáveis são numéricas e este algoritmo sabe trabalhar apenas com valores categóricos, as variáveis foram divididas em duas categorias (alto e baixo) com um ponto de corte *crisp* entre elas. Este ponto de corte foi determinado com auxílio da literatura médica, disponível no site da Sociedade Brasileira de Diabetes. Rodou-se então este algoritmo aplicando-se pós-poda e conseguiu-se uma árvore capaz de classificar corretamente 193 casos (75,4% do total).

Este conjunto já foi usado extensivamente na literatura, usando-se os mais variados métodos, com resultados das mais diversas precisões. Os mais qualificados obtiveram precisão de 100% no conjunto de treinamento (característica típica dos algoritmos de árvores de decisão que não aplicam nenhum tipo de poda), mas nunca excederam 81% no conjunto de teste.

O resultado de 100% no conjunto de treinamento é esperado em certos métodos, como por exemplo árvores de decisão, pois esta é uma característica do algoritmo, especialmente

se não for usado nenhum tipo de poda. Entretanto, este tipo de precisão normalmente é conseguido à custa de uma baixa capacidade de extrapolação, como podemos ver, por exemplo, no trabalho de SUÁREZ (2001), que cria árvores de decisão e consegue resultados de cerca de 75% de acerto. Outros algoritmos usaram árvores de decisão e chegaram a resultados semelhantes.

EGGERMONT *et al.* (2004b) comparam árvores de decisão usando algoritmo C4.5 com uma classificação baseada em programação genética. Esta última obtém resultados superiores, mas sua taxa de erro também é bastante alta (cerca de 26%).

BLAVYAS *et al.* (2001) utiliza métodos de interpolação usando técnica de treinamento de "leave-one-out", isto é, faz 768 treinamentos (tamanho do conjunto de dados completo) usando 767 dados para treino e um para validação. Este método é extremamente veloz, executando em frações de segundo, e conseguiu um desempenho de 76% de acerto em todos os dados – posto que o processo de "leave-one-out" é repetido para todos os elementos, não existe mais distinção entre conjunto de dados e conjunto de treinamento. É importante ter em mente que, dado o seu método de treinamento, este algoritmo conhece todas as instâncias do conjunto de dados e não existe nenhum tipo de extrapolação real.

Outro método extremamente veloz é a classificação baseada em Support Vector Machines. MANGASARIAN *et al.* (1999) usa este método para classificar os dados da diabetes, conseguindo resultados em torno de 78% de acerto com tempo de treinamento na faixa dos segundos. Outro artigo que utiliza SVM para classificação é (MEYER *et al*., 2002), mas este também não consegue resultados muito melhores, obtendo uma taxa de acerto de aproximadamente 77%.

LEE (2000) relata um método baseado em redes neurais que consegue uma taxa de acerto de aproximadamente 79% no conjunto. Este método baseia-se em técnicas bayesianas para escolha de modelo e consegue um dos melhores resultados descritos na literatura, apesar de ter um tempo de treinamento muito lento, quando comparado com outros métodos, tais com SVM ou métodos de regressão.

ALBA *et al.* (2004), compara a utilização de algoritmos genéticos para treinar redes neurais que utilizam métodos de gradiente descendente com os métodos tradicionais de treinamento destas redes e aplica este método em vários problemas de classificação, entre os quais o problema da diabetes. O artigo reporta resultados inferiores aos apontados até

agora, com taxas de acerto próximas de 72% nos melhores resultados, abaixo inclusive do previsor naïve usado como base de referência.

Foi aplicado então a algoritmo aqui proposto. Foram realizadas 100 rodadas do algoritmo, cada uma das quais em dois computadores simultaneamente, usando-se o algoritmo paralelo discutido no capítulo 3 desta tese. Cada computador executou até 80 gerações com 100 indivíduos cada.

Os melhores resultados foram obtidos quando se dividiu o universo de discurso das variáveis em cinco conjuntos fuzzy criados a partir da divisão uniforme do universo de discurso de cada variável, além de terem sido usados os operadores min-max para a composição fuzzy.

Os operadores genéticos tinham aptidão variável, com fitness inicial de 95% para o operador de crossover e variação linearmente decrescente para a mesma. O operador de mutação usado tinha uma probabilidade de 5% de alterar uma árvore quando aplicado.

Foi usada uma estratégia elitista no módulo de população, com a manutenção dois dois melhores indivíduos de uma população para a outra, de um total de 120 indivíduos. A execução de cada rodada foi interrrompida depois de 60 gerações ou depois de haver estabilização da avaliação do melhor indivíduo durante 10 gerações, o que poderia caracterizar convergência genética.

O melhor indivíduo obtido é caracterizado pela regra dada na tabela 6-2

SE [Alta(Massa_Corporal) AND Alta(Número_Gravidez)] OU [Muito_Alta(Massa_Corporal) AND NOT Baixa(Pressão)] OU [Alta(Concentração_Glicose)] ENTÃO Diabético

Tabela 6-2: Melhor regra para separação do conjunto dos índios diabéticos

Esta regra conseguiu um índice de classificação correta de 87% (443 classificações corretas em 512 instâncias) no conjunto de treinamento e 80% de acerto (205 acertos em 256 exemplos) no conjunto de teste.

Os resultados podem ser melhor compreendidos através da matriz de confusão, mostrada na tabela 6-3.

	Pessoa Sem Diabetes (155 casos)	Pessoa Com Diabetes (101 casos)
Previsto Sem Diabetes	129	25
Previsto Com Diabetes	26	76

Tabela 6-3: Matriz de confusão para os resultados obtidos pela aplicação do algoritmo proposto à base de dados da diabetes.

Estes resultados apontam para uma sensitividade de 83,2%, uma precisão de 83,7% e uma especificade de 75,1%, demonstrando uma preferência pelo conjunto mais numeroso, o que sugere a aplicação de uma técnica de compensação de desbalanceamento, questão que é discutida na seção de conclusão deste capítulo.

Algoritmo	Precisão (%)	U-Theil (Naïve)	U-Theil (ID3)
K Nearest Neighbour	69,0	1,148	1,585
Previsor Naïve (alta taxa de glicose igual a diabetes)	**73,0**	**1,000**	**1,097**
Previsor Naïve (baseado na heurística ID3)	**75,4**	**0,911**	**1,000**
Técnica de Interpolação	76,2	0,881	0,967
Splines	77,4	0,837	0,878
SVM	78,1	0,811	0,890
Redes Neurais + Algoritmo Evolucionário	78,2	0,807	0,886
Discriminante Linear	79,8	0,748	0,821
Método Proposto	**80,0**	**0,741**	0,796
Redes Neurais RBF + PCA	82,0	0,667	0,732

Tabela 6-4: Resultados de vários métodos de classificação aplicados ao conjunto da diabetes. O método proposto aqui na tese só perde para o método de redes neurais associado à análise de componentes principais. Note-se que o previsor naïve supera um dos métodos descritos na literatura.

Estes resultados são equivalentes aos melhores resultados obtidos na literatura e, apesar do tempo de execução do algoritmo aqui proposto ser muito maior do que outros algoritmos que obtêm resultados similares, como os métodos de SVM. Um ponto importante é que o método em questão possui um coeficiente U-Theil baixo (abaixo de 0,75), indicando que ele é um classificador bem mais poderoso que o previsor naïve proposto. Um resumo das taxas de acerto dos diferentes métodos é mostrada na tabela 6-4.

Uma forte qualidade deste resultado é sua simplicidade e a capacidade de interpretação direta dos resultados. Por exemplo, observando a regra acima, pode-se ver uma relação direta entre a massa corporal e a presença de diabetes tanto diretamente (através da presença dos conjuntos Alto e Muito_Alto para a variável Massa_Corporal) quanto indiretamente (no caso de um grande número de gestações, que normalmente implica em uma maior massa corporal da mulher). Este resultado é amplamente conhecido na medicina, podendo ser encontrado em vários textos médicos, como aqueles encontrados no site da Sociedade Brasileira de Diabetes e sua descoberta pelo algoritmo indica sua capacidade de extrair informação significativa de uma massa de dados, oferecendo-a de forma simplificada e compreensível para o destinatário final (usuário).

A falta de precisão de todos os métodos é patente – o melhor resultado tem taxas de erros próximas a 20% - o que poderia servir de estímulo para que outros pesquisadores busquem novos métodos capazes de obter melhores resultados. Entretanto, isto pode ser inviável nesta massa de dados.

É possível que a imprecisão apresentada por todos os métodos seja devida ao fato de que apesar da maioria das variáveis envolvidas neste teste serem relevantes para a compreensão desta síndrome metabólica, existem várias outras componentes envolvidas que não são citadas, tais como predisposição genética e quantidade de exercício realizada, entre outras. Todos estes importantes fatores estão ausentes do conjunto de dados disponível para estudos.

Além disto, temos o problema de que na prática, algumas das variáveis explicativas no conjunto de dados apresentam forte correlação. Por exemplo, estudos mostraram que indivíduos obesos, apresentando índice de massa corporal superior a 30, com diabetes ou não, apresentam, em jejum, níveis de insulina aumentados e a liberam mais após sobrecarga oral de glicose (SOCIEDADE BRASILEIRA DE DIABETES, 2004). Isto implica que na

prática, estas duas variáveis não são independentes e qualquer algoritmo que procure tratá-las como tal terá maior dificuldade em realizar uma classificação eficiente deste conjunto de dados.

A soma de todos estes fatores não implicam na desqualificação do método proposto aqui, tendo em vista sua capacidade de extrair de uma massa de dados conhecimento compreensível para o usuário. Isto faz com que, mesmo que a precisão da classificação do algoritmo aqui proposto não seja próxima de 100%, os resultados obtidos através de sua aplicação ainda são utilizáveis por um usuário (no caso específico desta aplicação, um médico) como um guia para melhor compreensão do fenômeno subjacente aos dados.

A.5 Aplicações do algoritmo proposto a dados categóricos

A.5.1 Análise de crédito

Este conjunto de dados, amplamente utilizado na literatura de classificação e conhecido popularmente como German Credit, também está disponível publicamente no repositório da UC-Irvine, consistindo em um conjunto de dados de análise de crédito de indivíduos alemães com base em seus dados financeiros/empregatícios.

O conjunto contém 1000 instâncias, cada qual com 20 diferentes atributos, variando desde dados contínuos (como idade e número de créditos com o banco) até dados categóricos (como a propriedade onde mora e tipo de emprego que a pessoa em questão possui).

Os atributos destes dados são os seguintes:

Status da conta existente: categórico, podendo assumir os valores { no-account, ODM, less-200DM, over-200DM}

- Duração da conta, em meses.
- Histórico de créditos: categórico, podendo assumir os valores {all-paid-duly, duly-till-now, bank-paid-duly, delay, critical}
- Propósito do empréstimo: categórico, podendo assumir os valores {car (new), car (used), furniture/equipment, radio/television, domestic appliances, repairs, education, business, others
- Valor do crédito, em marcos.
- Posses: categórico, podendo assumir os valores {less-100, less-500, less-1000, more-1000, unknown}
- Estado empregatício atual: categórico, podendo assumir os valores { unemployed, less-1yr, less-4yr, less-7yr, more-7yrs}
- Percentagem da prestação sobre valor da renda.
- Estado civil/sexo: categórico, podendo assumir os valores {male: divorced/separated, female: divorced/separated/married, male: single, male: married/widowed, female: single}
- Outras garantias: categórico, podendo assumir os valores {none, co-applicant, guarantor}
- Anos de residência.
- Propriedades: categórico, podendo assumir os valores {real estate, building society savings agreement, life insurance, car or other, not in attribute 6, unknown / no property
- Idade, em anos.
- Outras prestações: categórico, podendo assumir os valores {bank, stores, none}
- Moradia: categórico, podendo assumir os valores {rent, own, free}
- Número de créditos existentes no banco.
- Emprego: categórico, podendo assumir os valores {unemployed/ unskilled - non-resident, unskilled – resident, skilled employee / official, management/ self-employed, highly qualified employee/ officer}
- Número de dependentes.

- Possui telefone: categórico, podendo assumir os valores {no, yes}
- Estrangeiro: categórico, podendo assumir os valores {yes, no}

Os indivíduos podem ser classificados em duas classes (dignos de crédito ou não dignos de crédito). Assim como nos casos anteriores relatados nesta tese, estes atributos foram transformados em uma variável binária para classificação.

O conjunto foi dividido em duas partes: um conjunto de treinamento com 666 instâncias e um conjunto de avaliação com 334 instâncias. O conjunto de treinamento contém 209 pessoas (31,4%) que devem ter o crédito negado e 457 (68,6%) de pessoas merecedoras de crédito. No conjunto de testes esta proporção é de 243 (72,8%) de bons pagadores e 91 (27,2%) de pessoas a quem o crédito não pode ser concedido.

Este conjunto de dados foi escolhido para avaliação do algoritmo proposto nesta tese, pois, assim como o conjunto da diabetes, apresenta uma complexa relação entre os dados e os conjuntos a serem obtidas que não foi capturada de forma perfeita nos artigos pesquisados para esta tese.

EGGERMONT *et al.* (2004b) propõe uma abordagem baseada em programação genética que obtém uma taxa de acerto de apenas 74%. Entretanto, este artigo afirma que sua abordagem é melhor do que os algoritmos de árvore de decisão tradicionais, uma vez que sua implementação consegue uma precisão de apenas 72%.

AUER *et al.* (2002), ONODA *et al.* (2001) e GESTEL *et al.* (2001) aplicam máquinas de vetor de suporte, sendo que todos apresentam taxas de acerto em torno de 77% para o conjunto de avaliação.

Outras técnicas aplicadas não conseguem resultados mais impressionantes. Por exemplo, BAESENS *et al.* (2002) utiliza um previsor baseado em redes bayesianas aprendidas através do método de Monte Carlo e consegue uma precisão de 73,1%. No mesmo artigo é citado que um método bayesiano naïve consegue 76,7% de precisão no conjunto de treinamento.

O'DEA *et al.* (2001) descreve um algoritmo que combina características de teoria da informação para seleção de atributos com redes neurais e que apresenta uma performance igualmente pobre, atingindo uma precisão máxima no conjunto de treinamento igual 75,9%.

ELKAN (2001) propõe uma forma de avaliação de qualidade da solução a técnica adotada em muitos artigos, que consiste em associar uma matriz de custo a cada decisão tomada, e o resultado final é a média dos custos obtidos. Esta matriz de custo associa um custo a cada classificação correta ou incorreta, tendo um formato similar à matirz de confusão mostrada anteriormente, Para um problema de classificação de duas classes, a matriz de custo seria como visto na tabela 6-5, supondo que as classes se chamam "positivo" e "negativo".

	Classe Negativa	Classe Positiva
Previsto Negativo	C_{00}	C_{01}
Previsto Positivo	C_{10}	C_{11}

Tabela 6-5: Custos obtidos para cada uma das previsões associadas aos valores reais das classes.

A maioria dos artigos que usam esta métrica associaram os seguintes custos às previsões:

- $C_{00} = 0$
- $C_{11} = 0$
- $C_{10} = 5$
- $C_{01} = 1$

Estes números são baseados no fato de que para um banco, o custo de uma oportunidade perdida (negar crédito para uma pessoa que o merece) é muito menor que o custo do dinheiro perdido (oferecer um empréstimo para uma pessoa que provavelmente não o pagará). Em contrapartida negar crédito para uma pessoa que não é merecedora ou oferecer crédito para um bom pagador são decisões que não implicam em "custo" para a instituição de crédito, pois na realidade, convertem-se em lucro ao final do processo.

Para obter dados para inicializar a população mantida pelo programa, assim como para criar uma base de comparação para o desempenho do algoritmo aqui proposto, foi implementada uma versão do algoritmo ID3 para montagem de árvores de decisão. Como explicado anteriormente, este algoritmo é uma heurística do tipo "greedy" que se baseia na teoria da informação, tentando escolher um atributo de classificação que minimize a entropia associada à variável de interesse.

Fez-se a implementação usando-se um algoritmo de pós-poda para diminuir o tamanho da árvore e melhorar sua generalização. Os resultados obtidos para esta implementação apontam uma taxa de acerto de 87,8% na populaçao de treinamento e 76,7% na populaçao de teste, enquanto que de acordo com a matriz descrita acima, o custo associado a esta árvore é de 0,56. A árvore obtida neste treinamento está mostrada na figura 6-6.

É importante ter em mente que um previsor naïve que afirmasse que todas as pessoas sao dignas de crédito (categoria=good), obteria uma taxa de acerto, conforme a descrição do conjunto fornecida acima, de 68,6% na populaçao de teste e de 72,8% na população de treinamento.

O resultado obtido é perfeitamente consistente com o fato de que a maioria da população (especialmente da Alemanha, de onde estes dados são originários, e onde a população é famosa pelo seu respeito às leis e instituições) é boa pagadora. Assim, pode-se usar estes números como base de comparação para se estabelecer um desempenho mínimo aceitável.

Se for usado um previsor naïve que indique que todos os aplicantes não são dignos de crédito, o banco perderá o negócio em 72,8% dos casos, gerando um custo de 0,728 de acordo com a matriz de custo proposta acima.

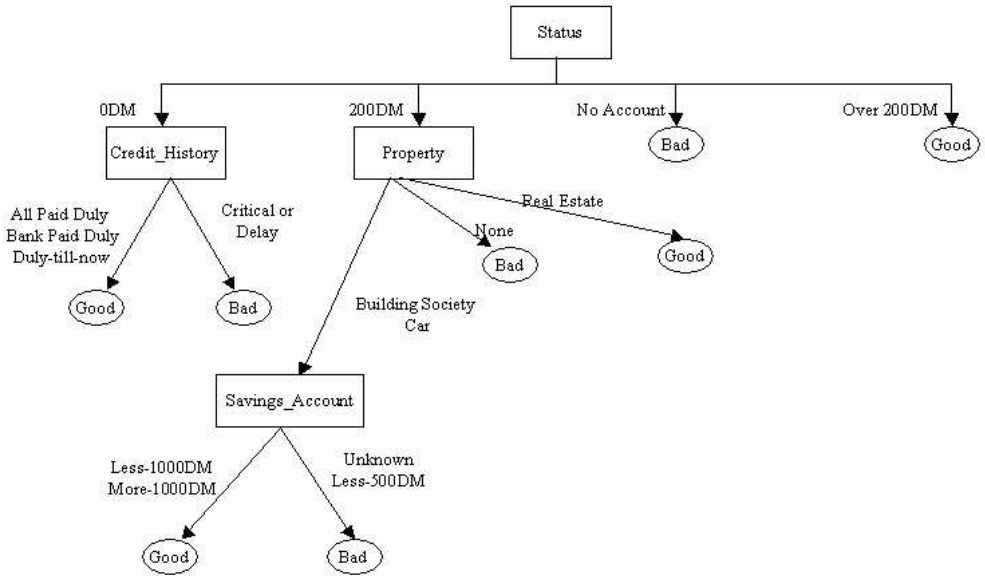


Figura 6-6 → Árvore de decisão calculada para o conjunto de dados German Credit. O desenho foi simplificado, colocando-se múltiplos valores por ramos, por questões de tamanho.

A árvore de decisão determinada pelo algoritmo ID3 foi transformada em regras e usada como inicialização para o algoritmo proposto nesta tese. Todas as regras colocadas foram definidas como desejáveis, de forma que o algoritmo daria preferência a estas, tendo a liberdade de omiti-las das regras finais.

Foram feitas 50 execuções do algoritmo de 80 gerações com 100 indivíduos cada, usando-se elitismo (preservação dos dois melhores elementos de cada geração para a próxima) e operador de crossover com probabilidade de uso linearmente decrescente. O operador de mutação, quando selecionado, foi aplicado com probabilidade de 5%. Para as variáveis numéricas foram usados três conjuntos (Alto, Médio e Baixo).

Criaram-se dois conjuntos para previsão da recorrência ("crédito bom" e "crédito ruim") e o número de regras médio por conjunto foi três. A inicialização do método iterativo de separação de dados categóricos (STIRR) foi feita de forma uniforme e foi usado o operador de soma para atualização dos pesos.

As regras obtidas pelo algoritmo foram aquelas descritas na tabela 6-6 a seguir.

➢ SE job IN {management,skilled} AND Baixo{Installments} ENTÃO Crédito Bom
➢ Se Status IN {over-200DM} ENTÃO Crédito Bom
➢ Se Credit_History IN { All Paid Duly , Bank Paid Duly, Duly-till-now} ENTÃO Crédito Bom
➢ SE Baixa(Age) AND Housing IN {rent,free} ENTÃO Crédito Ruim
➢ SE Property IN {None} AND Personal_Status IN {Female_Divorced,Male_Divorced,Single_Male} ENTÃO Crédito_Ruim

Tabela 6-6: Regras que descrevem a separação do conjunto de pessoas com bom e mau crédito.

Estas regras atingem uma taxa de acerto de 86% no conjunto de treinamento (573 acertos em 666 exemplos) e de 78,4% no conjunto de teste (262 acertos em 334 exemplos). O custo obtido para esta solução de acordo com a métrica descrita acima é de 0,54.

O repositório da UC-Irvine lista vários resultados que foram conseguidos na literatura de classificação usando a matriz de custo como métrica de precisão. As fontes não são mencionadas para os resultados, mas eles são dignos de nota, podendo ser vistos na

tabela 6-7, junto com sua comparação com os métodos naïve e os resultados obtidos pela aplicação do algoritmo proposto nesta tese.

A utilização das regras do ID3 é interessante posto que exemplifica o acréscimo de regras determinadas com base em conhecimento anterior. Outras regras que fossem proveniente de especialistas na área poderiam ter sido fornecidas ao algoritmo.

A melhora obtida pelo algoritmo proposto em relação ao método ID3 é pequena, mas mesmo uma diferença de custo pode justificar a aplicação de um algoritmo aos olhos de um usuário final (banqueiro), dado o prospectivo aumento de ganhos.

Algoritmo	Custo	U-Theil (Sobre ID3)	U-Theil (Sobre Naïve)
Discriminante Linear	0.535	0.955	0,735
Algoritmo Proposto	**0.54**	**0.964**	**0,741**
ID3	**0.56**	**1.000**	**0.769**
Cart	0.613	1.094	0.842
K Nearest Neighbour (KNN)	0.694	1.239	0.953
Redes Bayesianas	0.703	1.255	0,965
Previsor Naïve (Negar Crédito para todos)	**0.728**	**1.300**	**1.000**
Redes Neurais BackPropagation	0.772	1.378	1.060
Árvores Bayesianas	0.778	1.389	1.068
Funções de Base Radial	0.971	1.733	1.334
C4.5	0.985	1.758	1.353
Redes Neurais de Kohonen	1.160	2.071	1.593

Tabela 6-7: Resultados de vários métodos de classificação, conforme descrito no repositório do UCI, comparado com o previsor naïve, a árvore de decisão montada usando-se o algoritmo de ID3 e o resultado do algoritmo proposto aqui nesta tese.

Como em outras aplicações, existe um método que tem um desempenho superior àquele exibido pelo algoritmo aqui apresentado. Entretanto, o método ganhador não é sempre o mesmo e o algoritmo proposto sempre se apresenta com um desempenho entre

bom e ótimo, o que permite concluir que ele é um classificador genérico mais eficiente que a média.

Outro ponto a ser considerado é a questão da usabilidade dos resultados. A análise por discriminante linear, o método que obteve o melhor desempenho neste exemplo, usa uma definição de um hiperplano separador que pode ser pouco compreensível para um banqueiro.

Por outro lado, os resultados obtidos pelo algoritmo proposto, dado que usam uma definição lingüística próxima àquela usada no cotidiano, criam uma compreensão do fenômeno subjacente à questão do não pagamento, podendo levar a uma maior usabilidade dos seus resultados por parte do usuário final.

A.5.2 Câncer no seio

O conjunto de dados do câncer do seio de Ljubljana também foi obtido no repositório da UC-Irvine, sendo originário do Centro Médio Universitário de Oncologia, de Ljubljana, antiga Iugoslávia.

Este conjunto de dados difere do conjunto normalmente denominado de Câncer do Seio de Wisconsin, pois está focado em dados do paciente, enquanto que o segundo conjunto consiste em uma série de dados citopatológicos. O conjunto de Wisconsin também foi motivo de estudos extensos e foi modelado com sucesso em vários trabalhos, entre os quais podemos apontar (MANGASARIAN *et al.*, 1999).

O conjunto de Ljubljana inclui 201 instâncias de pacientes com câncer sem recorrência e 85 de pacientes cujo câncer retornou em até um ano após o diagnóstico e tratamento iniciais.

As instâncias são descritas pelos seguintes 9 atributos:

- Idade: categórico, podendo assumir os valores {10-19, 20-29, 30-39, 40-49, 50-59, 60-69, 70-79, 80-89, 90-99}
- Menopausa: categórico, podendo assumir os valores {lt40, ge40, premeno}

- Tamanho do tumor: categórico, podendo assumir os valores {0-4, 5-9, 10-14, 15-19, 20-24, 25-29, 30-34, 35-39, 40-44, 45-49, 50-54, 55-59}
- Nós invadidos : categórico, podendo assumir os valores {0-2, 3-5, 6-8, 9-11, 12-14, 15-17, 18-20, 21-23, 24-26, 27-29, 30-32, 33-35, 36-39}
- Encapsulado: categórico, podendo assumir os valores {yes, no.}
- Grau de malignidade, numérico assumindo valores de 1 a 3.
- Seio afetado: categórico, podendo assumir os valores {left, right}
- Quadrante do seio afetado: categórico, podendo assumir os valores {left-up, left-low, right-up, right-low, central}
- Irradiado: categórico, podendo assumir os valores {yes, no}

Todos os atributos são nominais, com exceção do grau de malignidade, que consiste em um número variando de 1 a 3. Este atributo poderia ser tratado também como uma classe nominal (baixa, média e alta taxa de malignidade), mas se isto fosse feito perder-se-ia a ordenação entre os conceitos.

O conjunto de treinamento consiste em 191 instâncias, 116 (60,6%) das quais de pacientes sem recorrência e 75 (39,45%) de pacientes cujo câncer retornou. No conjunto de teste, temos 95 exemplos, 70 (73,7%) dos quais de pacientes que não recorreram e 25 (26,3%) de pacientes com recorrência.

Um previsor naïve simples que serve como base de comparação para o desempenho de qualquer previsor consiste naquele que prevê que todos os pacientes não sofrerão recorrência do câncer, obtendo uma taxa de sucesso de 73,7% no conjunto de treinamento.

Vários trabalhos antigos tentaram fazer previsões com base neste conjunto, com resultados próximos ou inferiores ao previsor naïve. MICHALSKI *et al.* (1986) obteve uma precisão na faixa de 66% a 72%, CLARK *et al.* (1987), de 65% a 72% e TAN *et al* (1988) de 68% a 73,5%.

DUCH *et al.* (2001) usaram redes neurais para extração de regras e obteve um resultado de 77,1% no conjunto de treinamento. Um previsor naïve Bayesiano, descrito no mesmo artigo, obteveve uma precisão de 75,9% no mesmo conjunto.

Todos estes desempenhos são similares ao apresentado pelo previsor naïve escolhido. O sistema baseado em redes neurais, que teve o melhor desempenho entre todos os trabalhos descritos acima, obteve um coeficiente U modificado de 0,871.

Alguns trabalhos mais recentes baseados em máquinas de vetor de suporte (SVM), como (AUER *et al.*, 2002, ONODA *et al.*, 2001) obtiveram resultados superiores, chegando a uma precisão próxima de 97%.

Uma implementação direta do algoritmo ID3 usando pós-poda e considerando todos os atributos como nominais, obteve uma taxa de classificação correta em 84% dos casos do conjunto de treinamento e 77% dos casos do conjunto de validação. As variáveis que ficaram nos pontos mais altos na árvore de treinamento, foram o tamanho do tumor, o número de nós invadidos e a malignidade do tumor. Estas três variáveis confirmam o bom senso médico e sugerem que um médico poderia obter resultados melhores do que a maioria dos métodos propostos.

Com base nesta premissa, pediu-se a um oncologista que analisasse os dados e predizesse a recorrência do câncer no conjunto de teste. O médico criticou as variáveis disponíveis, afirmando que elas não lhe forneciam todas as informações necessárias para avaliar corretamente os pacientes.

Mesmo sob estas restrições, o médico predisse corretamente a recorrência do câncer em 86% dos pacientes. Este desempenho é superior àquele obtido pela maioria dos métodos automáticos e sugere que, na posse de todos os dados que considera necessários, o médico poderia obter resultados mais próximos daqueles obtidos pelos melhores métodos de classificação.

Levando-se em consideração estas restrições apontadas pelo médico e os resultados oferecidos pelo algoritmo ID3, foi então aplicado o algoritmo proposto nesta tese.

Foram feitas 30 execuções do algoritmo de 60 gerações com 100 indivíduos cada, usando-se elitismo (preservação dos dois melhores elementos de cada geração para a próxima) e operador de crossover com probabilidade de uso linearmente decrescente. O operador de mutação, quando selecionado, foi aplicado com probabilidade de 10%.

Considerou-se a variável malignidade como sendo numérica. Apesar desta só poder assumir um conjunto fechado de valores (1,2 e 3), estes estão ordenados entre si e a transformação em variável numérica fornece a informação de a malignidade de grau 1 está

mais próxima da de grau 2 do que da de grau 3, Esta variável foi dividida em 3 conjuntos fuzzy (baixa, média e alta), cada um dos quais com pico em um dos valores que a variável pode assumir.

Criaram-se dois conjuntos para previsão da recorrência ("paciente recorrente" e "paciente não recorrente") e o número de regras médio por conjunto foi dois. Para demonstrar a capacidade de embutir conhecimento externo no algoritmo, as seguintes regras foram sugeridas para o algoritmo. Para o conjunto de Não recorrentes, sugeriu-se que o conjunto de Baixa(Malignidade) teria influência positiva enquanto que para o conjunto de recorrentes sugeriu-se que se incluísse o conjunto de Alta(Malignidade) nas regras.

A inicialização do método iterativo de separação de dados categóricos (STIRR) foi feita de forma uniforme e foi usado o operador de soma para atualização dos pesos.

Os operadores fuzzy usados foram do tipo min-max

A implementação do algoritmo proposto nesta tese retornou as regras para classificação dos pacientes descritas na tabela 6-8.

- SE Tamanho IN {0-4,5-9,10-14,15-19,20-24} ENTÃO Paciente não recorrente
- SE Irradiado IN {yes} AND Baixa(Malignidade} ENTÃO Paciente não recorrente
- SE Nó_Encapsulado in {no}ENTÃO Paciente não recorrente
- SE Tamanho IN {25-29,30-34,35-39,40-44} AND Alta(Malignidade) ENTÃO Paciente recorrente

Tabela 6-8: Regras que separam o conjunto de pacientes que terão recorrência do câncer daqueles que não terão, conforme descoberto pelo algoritmo proposto nesta tese.

Este conjunto de regras simples previu corretamente a recorrência de 87 em 95 pacientes (91,6%) do conjunto de teste obtendo um coeficiente U Modificado de 0,319. Este desempenho é consideravelmente superior ao previsor naïve e melhor do que todos os métodos apresentados até agora, com exceção das máquinas de vetor de suporte (SVM), sendo superior também ao desempenho conseguido pelo oncologista consultado.

Entretanto, a clareza da informação oferecida por estas regras é um fator adicional a se considerar em termos de usabilidade de resultados. Um médico poderia facilmente usar as informações fornecidas por estas regras como base de um projeto de estudo sobre prognóstico de pacientes com câncer no seio. O mesmo não pode ser dito por resultados obtidos através do uso de SVM.

Este conjunto de dados possui características impotantes que demonstram o fato de que a maioria dos métodos simplesmente ataca os dados sem considerar o campo de onde eles se originaram, a medicina.

Existe ao menos uma variável espúria neste conjunto de dados que consiste no seio em que o câncer se desenvolveu. Um oncologista consultado declarou não haver qualquer motivo para que se considere o seio canceroso como variável relevante para o processo de prognóstico do paciente.

Este fato é comprovável empiricamente nos dados. O número de pacientes que desenvolveu câncer no seio esquerdo consiste em 51% dos dados, contra 49% do seio esquerdo. O ganho de informação (IG) associado à escolha desta variável, definido pela fórmula:

$$IG(\text{Pr}\,ognóstico) = H(\,prognóstico) - H(\,prognóstico \mid seio)\,, \text{ onde}$$

- H(prognóstico) é a entropia associada à variável prognóstico
- H(prognóstico|seio) é a entropia condicional da variável prognóstico em relação à variável seio, valor este que é dado pela seguinte fórmula:

$$H(\,prognóstico \mid seio) = \sum_j p(seio = v_j) H(\,prognóstico \mid seio = v_j)$$

O ganho de informação obtido é igual a 0,005, indicando que não há nenhum valor classificatório relevante associado a esta variável. Todas as outras variáveis do conjunto, com exceção do quadrante do seio onde o câncer ocorreu, possuem ganhos de informação pelo menos 10 vezes maior do que esta variável. Isto indica que qualquer classificador que se utilize desta variável, obterá valores espúrios decorrentes apenas da utilização de um pequeno espaço amostral.

Ignorar a ciência originadora de um determinado conjunto de dados é prática comum dentro da literatura de classificação, mas isto não faz com que esta prática seja elogiável. Ao contrário, ela prejudica o desempenho da maioria dos métodos ou então faz

com que estes produzam resultados com pouca relevância para o usuário final dos dados (no caso desta base de dados específica, os médicos).

Por isto, é importante a existência de métodos que permitam embutir conhecimento pré-existente e a utilização de técnicas de comparação dos métodos utilizados com métodos adequados à ciência em questão.

A.6 Comentários gerais sobre a aplicação de classificação

Muitos trabalhos justificam a aplicação de seus algoritmos, especialmente aqueles baseados em interpolação de funções, com base em questões de velocidade, alegando que, mesmo que eles não tenham uma precisão maior do que os trabalhos previamente estabelecidos, eles são mais velozes. Com base nesta linha de argumentação, a qualidade da informação é menos importante do que a velocidade com a qual ela foi obtida, o que certamente é falacioso.

O algoritmo proposto nesta tese é inerentemente mais lento, posto que é baseado em um algoritmo evolucionário (EA). Por serem heurísticas, EAs normalmente necessitam múltiplas execuções, o que faz com que o tempo requerido para a obtenção de resultados seja necessariamente maior do que aquele em algoritmos velozes. Entretanto, isto não significa por si só que este algoritmo seja inferior.

As aplicações apontadas para o algoritmo desta tese não são “on-line” ou de tempo real. Apontou-se aqui o uso deste algoritmo em aplicações que podem ser caracterizadas como parte da área e mineração de dados, na qual busca-se extrair informação valiosa de um determinado conjunto de dados. Isto posto, fica claro que é necessário se privilegiar a qualidade da informação, não só em termos da adequação do modelo aos dados existentes quanto da clareza e da usabilidade destes dados para a pessoa que será o destinatário final desta informação.

Esta clareza está presente no algoritmo proposto nesta tese. As variáveis de interesse e as características básicas de seu relacionamento estão claramente presentes em todos os resultados oferecidos pelo algoritmo, o que permite que estes resultados sejam a base de um processo decisório, seja como hipótese ou como modelo final proposto.

É claro que a velocidade é uma característica importante, especialmente se for desejável analisar bases de dados de grande volume (VLDB). Entretanto, existem trabalhos na área de mineração de dados, tais como (TOIVONEN, 1996) que usam amostras aleatórias de grandes bancos de dados como os conjuntos a serem analisados pelos seus algoritmos, de forma que o tamanho da massa de dados seja tratável por um algoritmo mais lento. Este tipo de abordagem poderia ser indicado para se extender a aplicabilidade do algoritmo proposto aqui para bases de dados de grande tamanho, aliado a uma estratégia de otimização do algoritmo, descrita em mais detalhes no capítulo final desta tese.

Em termos de precisão, o algoritmo proposto nesta tese apresentou resultados inferiores àqueles encontrados nos trabalhos com máquinas de vetor de suporte (SVM), que é a técnica que geralmente apresenta os melhores resultados. O algoritmo aqui proposto, entretanto, tem um desempenho bom, superando a maioria das outras técnicas descritas na literatura de classificação.

É importante ressaltar, entretanto, que o algoritmo proposto não foi projetado com o intuito de classificação, e, por isso, o desempenho razoável no problema de classificação é um bônus, mostrando a possibilidade de se adaptar o algoritmo para diversos problemas.

Um problema dos algoritmos de SVM é o fato de ainda não ter sido estudado de forma intensiva a incorporação de conhecimento pré-existente de forma a restringir o espaço de busca. Isto poderia ser feito incorporando-se outras condições limitantes do espaço de busca, mas esta questão começou a ser estudada apenas recentemente, e ainda não está bem desenvolvida. Um exemplo de trabalho que realiza esta incorporação é (POZDNOUKOV *et al.*, 2004).

A incorporação de conhecimento foi analisada e efetuada dentro do trabalho aqui proposto, especialmente nos trabalhos de bioinformática. Sem um mecanismo para fazer esta incorporação, a aplicabilidade de um algoritmo fica reduzida.

Outra questão importante apontada neste capítulo é a de se analisar os resultados obtidos à luz das características inerentes aos conjuntos de dados utilizados e da área científica de onde estes são provenientes.

Antes de se apontar uma determinada precisão como um sucesso, é necessário se analisar os dados, de forma a verificar se aquela precisão é significativamente melhor do que a que seria obtida com classificadores que se utilizam de métodos naïve, que serve como um limite mínimo de desempenho interessante.

Ademais, estudos da área científica que originou os dados podem permitir uma análise mais bem fundamentada dos dados, entendendo as limitações que processos classificatórios podem ter e permitindo que se analise *a priori* a necessidade de se efetivamente utilizar um algoritmo de classificação.

A.7 Conclusão

A aplicação do algoritmo a problemas de agrupamento apresentou resultados interessantes. O algoritmo foi aplicado a conjuntos de dados amplamente utilizados pelos pesquisadores mundialmente, e que têm se tornado benchmarks para novas técnicas.

Os resultados obtidos na aplicação do conjunto das flores íris demonstrou a capacidade do algoritmo de realizar um agrupamento de forma eficaz. Este conjunto é simples e linearmente separável, mas serve como uma demonstração da aplicabilidade da técnica.

Em seguida aplicou-se o algoritmo obtido ao conjunto de dados da diabetes dos índios Pima. Este conjunto se mostrou refratário a todos os principais algoritmos pesquisados durante esta tese, os quais nunca conseguiram resultados muito superiores a 80% de classificação correta.

O algoritmo aqui apresentado também manteve-se nesta faixa de precisão, a qual, como discutido, é possivelmente um limite para todos os algoritmos computacionais, visto que há forte correlação entre os dados apresentados e ausência de outros fatores que poderiam se mostrar expressivos. Como discutido, a precisão pode ser mascarada no caso

de desbalanceamento de classes, como é o caso deste conjunto de dados, dando preferência a classe predominante, o que sugere a aplicação de outras métricas, como sugerido em (ESPINDOLA, 2004).

A aplicação do algoritmo proposto a este conjunto demonstrou que este pode apresentar problemas para lidar com classes desbalanceadas, apresentando uma precisão mais alta que a especificidade. Isto faz com que seja interessante considerar, no futuro, a aplicação de alguma técnica de compensação de desbalanceamente, como a sugerida em (BATISTA *et al.*, 2004).

O principal benefício da aplicação deste algoritmo então consiste na simplicidade das regras obtidas que podem servir como guias simples para médicos que desejem compreender o fenômeno subjacente à doença que aflige seus pacientes, o que não é possível com os métodos tipo caixa preta já mencionados nesta seção.

Por último, porém não menos importante, este algoritmo é capaz de lidar com dados categóricos de forma direta, sem a necessidade de convertê-los para um formato numérico artificial.

No caso dos exemplos categóricos demonstrados nesta tese, o algoritmo aqui proposto obteve desempenho superior a todos os métodos naïve, sendo também superior ao especialista oncologista que analisou os dados do conjunto sobre câncer no seio.

Em ambas as aplicações, existe um método que tem um desempenho superior àquele exibido pelo algoritmo aqui apresentado. Entretanto, o método ganhador não é sempre o mesmo, apesar de ser claro que as máquinas de vetor de suporte têm um desempenho superior ao algoritmo aqui proposto.

Entretanto, o algoritmo proposto nesta tese sempre apresenta um desempenho entre bom e ótimo, o que permite concluir que ele é um classificador genérico eficiente e cujo uso pode ser considerado em várias aplicações distintas. Ademais, ele é mais eficiente que a maioria dos métodos existentes.

Outro ponto a ser considerado é a questão da usabilidade dos resultados. O método que parece ser o melhor em termos de classificação, inclusive em situações em que é necessária uma separação não linear, são as máquinas de vetor de suporte.

O problema deste método é que são necessárias transformações não lineares nos dados originais e o resultado obtido é apresentado na forma de um hiperplano separador no

espaço transformado, que pode ainda estar baseado em variáveis que não tenham um forte significado no processo formatório dos dados. Este tipo de resultado pode ser pouco compreensível para o usuário final da informação, que tende a ser um leigo em termos de matemática (um médico, um banqueiro, etc.).

Por outro lado, os resultados obtidos pelo algoritmo proposto, dado que usam uma definição lingüística próxima àquela usada no cotidiano, criam uma compreensão do fenômeno subjacente à divisão dos dados nos conjuntos propostos (dignos/indignos de crédito, diabéticos ou não, etc.), podendo levar a uma maior usabilidade dos seus resultados por parte do usuário final.

Uma conseqüência do trabalho realizado para esta tese foi a percepção de que é necessária uma análise profunda sobre as pesquisas realizadas na aplicação dos mais diversos algoritmos a problemas de classificação, em termos da necessidade de uma avaliação coerente da aplicabilidade das pesquisas realizadas e da qualidade dos resultados obtidos.

Quando preparando um trabalho sobre classificação, um pesquisador deve se perguntar se os resultados obtidos são realmente significativos, não importando a percentagem de acertos que o sistema em questão obtém. Isto é especialmente verdade tendo em vista que determinados patamares de acerto podem ser mais significativos em um problema do que em outro. Por exemplo, uma taxa de acerto em torno de 80% é péssima para o conjunto da íris, mas muito boa para o conjunto da diabetes.

Nesta tese foi proposto um coeficiente de U-Theil modificado associado ao uso de métodos naïve que embutem algum conhecimento e/ou análise dos dados. Esta métrica é extremamente simples, porém é valiosa na avaliação do valor real da precisão obtida por um método. Se esta métrica fosse aplicada à maioria dos trabalhos de classificação, a sua real eficência seria descoberta, em termos relativos a previsores simples que não exigem grandes esforços computacionais.

Esta métrica é apenas uma sugestão, não sendo o interesse final deste trabalho sugeri-la como padrão-ouro da avaliação de métodos de classificação. KEOGH *et al.* (2003) já apontaram a existência de um grande número de coeficientes de testes, muitos dos quais necessitam de maiores avaliações e comparações para com outros mais antigos e amplamente usados.

O coeficiente de U-Theil modificado foi usado nesta tese apenas como uma forma simples de criar um padrão de comparação para com métodos naïve. Entretanto, sua simplicidade e capacidade de fazer uma comparação direta fazem dele extremamente atrativo para usos em trabalhos de classificação.

O uso deste coeficiente e a análise realizada nesta tese levam à conclusão de que não é suficiente que os pesquisadores se atenham à apresentação de resultados absolutos ou que comparem seus resultados apenas com artigos anteriores. Os conjuntos de dados sendo analisados devem ser estudados de forma extensa e sua estrutura inerente deve ser apresentada para o leitor.

Além disto, a área científica que gerou o conjunto de dados em questão deve ser estudada para que o pesquisador tenha suficiente informação para compreender a validade e aplicabilidade de seus estudos.

Não é razoável simplesmente obter um conjunto de dados e iniciar um estudo diretamente, sem entender o contexto gerador dos números dentro deste conjunto. Por exemplo, um pesquisador que queira trabalhar com o conjunto de dados do câncer deve entender o que é esta doença e como as variáveis presentes no conjunto de dados afetam seu desenvolvimento.

Este tipo de análise foi feito nesta tese e, por conseguinte, os resultados obtidos ganharam uma significância maior. Com estes estudos, pôde-se entender certas limitações inerentes aos conjunto da diabetes e do câncer, além de prover uma discussão saudável sobre possíveis limitações de precisão no conjunto das írises, que é um representante interessante de conjuntos de dados provenientes de meios ambientes reais.

Outro ponto a ressaltar é a questão de que os resultados apresentados por este trabalho apresentam uma análise tentativa dos fenômenos subjacentes aos conjuntos de dados. Isto faz com que, mesmo que a precisão obtida não seja extremamente alta, o método aqui proposto sirva como uma ferramenta de extração de conhecimento e de apoio para pesquisadores em áreas diversas de conhecimento.