



MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

ALGORITMOS EVOLUTIVOS APLICADOS AO PROCESSO DE ANÁLISE DE
DADOS GEOGRÁFICOS

Adair Santa Catarina

INPE
São José dos Campos
2005

RESUMO

A utilização intensiva de computadores e de algoritmos computacionais na análise de dados espaciais fez surgir uma nova área de conhecimento: a geocomputação. Neste relatório abordou-se os conceitos elementares de geocomputação e modelagem de dados geográficos. Com maior ênfase estudou-se como os algoritmos evolutivos podem ser utilizados no processo de análise de dados geográficos, particularmente os algoritmos genéticos. Estudou-se três sistemas utilizados na construção de modelos de interação espacial: *model breeders*, GARP e DesktopGARP. Ao final do trabalho teceu-se algumas considerações finais, onde erros de implementação do GARP são discutidos e as soluções apresentadas. Também propôs-se a modificação dos operadores genéticos envolvidos no GARP e DesktopGARP, bem como alterações nos tipos de regras de decisão utilizadas em ambos os sistemas, e a hibridização do GARP através da inserção do algoritmo *Simulated Annealing*.

SUMÁRIO

LISTA DE FIGURAS	iii
1 INTRODUÇÃO.....	1
2 CONCEITOS GERAIS DE GEOCOMPUTAÇÃO.....	2
2.1 ELEMENTOS DOS SISTEMAS DE TRATAMENTO DA INFORMAÇÃO ESPACIAL.....	2
2.2 O PROCESSO DE ANÁLISE ESPACIAL	3
2.3 CONCEITOS BÁSICOS EM ANÁLISE ESPACIAL	4
3 MODELAGEM ESPACIAL	6
3.1 CONCEITO DE MODELO	6
3.2 MODELAGEM – ASPECTOS CONCEITUAIS	6
3.2.1 O Espaço e o Tempo.....	7
3.3 TIPOS DE MODELOS	9
4 ALGORITMOS GENÉTICOS (AGs)	11
4.1 INTRODUÇÃO AOS AGs	11
4.2 OS OPERADORES GENÉTICOS	14
4.2.1 Seleção por Monte Carlo.....	15
4.2.2 Elitismo.....	15
4.2.3 Cruzamento e Mutação	16
4.2.4 O Operador de Inversão.....	20
4.3 PARÂMETROS GENÉTICOS.....	20
4.4 HIBRIDIZAÇÃO	21
4.4.1 <i>Simulated Annealing (SA)</i>	22
4.4.2 O Algoritmo SA.....	24
4.4.3 Exemplo de Hibridização de AG com SA.....	25
5 APLICAÇÕES DE AGs NO PROCESSO DE ANÁLISE DE DADOS GEOGRÁFICOS..	26
5.1 CLASSES DE APLICAÇÕES DE AGs NO PROCESSO DE ANÁLISE DE DADOS GEOGRÁFICOS.....	26
5.1.1 Otimização em Redes	27
5.1.2 <i>Map Labeling</i>	28
5.1.3 Plano de Uso de Solos	29
5.1.4 Localização de Facilidades.....	30
5.1.5 Construção de Modelos de Interação Espacial.....	30
5.2 FERRAMENTAS “INTELIGENTES” PARA ANÁLISE DE DADOS GEOGRÁFICOS.....	31
5.2.1 <i>Model Breeders</i>	32

5.2.2 GARP <i>Modelling System – Genetic Algorithm for Rule Set Production</i>	34
5.2.2.1 Regras.....	35
5.2.2.2 Codificação das Regras.....	38
5.2.2.3 Mecanismo Evolutivo.....	38
5.2.3 DesktopGARP.....	41
5.2.3.1 Seleção do Melhor Subconjunto Solução.....	42
6 CONSIDERAÇÕES FINAIS	46
6.1 ERROS DE CÓDIGO.....	46
6.2 SUGESTÕES.....	48
REFERÊNCIAS BIBLIOGRÁFICAS	50

LISTA DE FIGURAS

FIGURA 3.1 – Tipos de modelos	10
FIGURA 4.1 – Fluxograma que descreve brevemente um algoritmo genético.	13
FIGURA 4.2 – Um exemplo de seleção através do algoritmo Monte Carlos ou método da roleta	15
FIGURA 4.3 – Um exemplo do operador de cruzamento em um ponto.	18
FIGURA 4.4 – O algoritmo SA	24
FIGURA 5.1 – Exemplo da codificação empregada no cromossomo.....	33
FIGURA 5.2 – Forma Geral de uma Regra.....	36
FIGURA 5.3 – Exemplo de Regra Atômica	36
FIGURA 5.4 – Exemplo de Regra BIOCLIM	37
FIGURA 5.5 – Exemplo de Regra de Faixas.....	37
FIGURA 5.6 – Exemplo de Regra Logística.....	37
FIGURA 5.7 – Conjunto de regras.....	38
FIGURA 5.8 – Exemplo de <i>crossover</i> sobre cromossomos binários	38
FIGURA 5.9 – Exemplo de operação cruzamento sobre as regras no GARP	39
FIGURA 5.10 – Exemplo de operação de junção sobre as regras no GARP	39
FIGURA 5.11 – Exemplo de operações de mutação sobre as regras no GARP	39
FIGURA 5.12 – O processo de seleção do GARP	41
FIGURA 5.13 – Interface do DesktopGARP	41
FIGURA 5.14 – Matriz de Confusão	42
FIGURA 5.15 – Curva padrão para a relação Omissão/Comissão.....	43
FIGURA 5.16 – Classificação das regiões quanto à intensidade dos erros de omissão/comissão	43
FIGURA 5.17 – Aplicação do limiar para eliminar modelos com muitos erros de omissão... ..	44
FIGURA 5.18 – Identificação da região com os melhores modelos ajustados	44
FIGURA 5.19 – Parâmetros para seleção do melhor subconjunto solução.....	45
FIGURA 6.1 – Trecho do código da rotina Mutate() – módulo heuristic.c.....	47
FIGURA 6.2 – Trecho do código da rotina Crossover() – módulo heuristic.c	47
FIGURA 6.3 – Trecho do código da rotina Join() – módulo heuristic.c.....	47
FIGURA 6.4 – Código corrigido – rotina Mutate() – módulo heuristic.c	47
FIGURA 6.5 – Código corrigido – rotina Crossover() – módulo heuristic.c.....	48
FIGURA 6.6 – Código corrigido – rotina Join() – módulo heuristic.c	48
FIGURA 6.7 – Exemplo de operação cruzamento sobre as regras no GARP	48
FIGURA 6.8 – Exemplo de cruzamento com operador aritmético por média	49

1 INTRODUÇÃO

A busca pelo conhecimento presente em conjuntos de dados geográficos, através do intensivo de computadores, fez surgir uma nova área de conhecimento: a Geocomputação.

Algoritmos “inteligentes” tem sido utilizados para explorar conjuntos de dados geográficos. Dentre estes algoritmos de busca “inteligentes” estão os Algoritmos Genéticos, utilizados na implementação de ferramentas como os *model breeders* e o GARP.

Model breeders são ferramentas capazes de encontrar relações entre variáveis independentes e dependentes. O GARP é uma ferramenta amplamente utilizada para modelagem de nichos ecológicos, ou seja, dado um conjunto amostra de uma determinada espécie e um conjunto de *layers* de informações ambientais e climáticas, o algoritmo é capaz de criar um modelo para prever quais regiões são aptas para o desenvolvimento daquela espécie.

Neste trabalho efetuou-se um estudo dos conceitos básicos de geocomputação e modelagem de dados geográficos. Com mais ênfase estudou-se como os algoritmos evolutivos, particularmente os algoritmos genéticos, podem ser utilizados no processo de análise de dados geográficos.

Uma revisão sobre algoritmos genéticos foi realizada, bem como um levantamento de diversos trabalhos que os relaciona com informações espaciais. Em particular estudou-se ferramentas como os *model breeders*, o GARP e o DesktopGARP.

Ao final do trabalho teceu-se algumas considerações finais sobre o assunto pesquisado. Também foram apontadas algumas falhas encontradas no programa GARP, e as devidas correções, no que tange ao módulo “inteligente” que cria o modelo de distribuição de espécies. Sugeriu-se ainda possíveis abordagens para modificação dos operadores genéticos e o conjunto de regras utilizados no sistema.

2 CONCEITOS GERAIS DE GEOCOMPUTAÇÃO

Geocomputação é um campo de pesquisa emergente que propõe o uso de técnicas intensivas em computação, tais como redes neurais, busca heurística e autômatos celulares para análise de dados espaciais (CÂMARA e MONTEIRO, 2001).

O termo geocomputação foi cunhado por OPENSHAW e ABRAHART (1996) para descrever o uso da computação intensiva na descoberta de conhecimento nas áreas de geografia humana e física, especialmente aqueles métodos não convencionais de agrupamento e análise.

Atualmente o termo inclui a análise de dados espaciais, modelagem dinâmica, visualização e os processos dinâmicos espaço-temporais (LONGLEY, 1998).

2.1 ELEMENTOS DOS SISTEMAS DE TRATAMENTO DA INFORMAÇÃO ESPACIAL

Os Sistemas de Informações Geográficas (SIGs) atuais possibilitam armazenar, organizar, recuperar e modificar informações sobre a distribuição espacial de recursos naturais, dados geo-demográficos, redes de utilidade pública e muitos outros tipos de dados localizados na superfície da terra. Um desafio para os próximos anos é transformar estes sistemas, essencialmente estáticos, em ferramentas capazes de prover representações realistas de processos espaço-temporais. A modelagem de grande quantidade de processos físicos, em aplicações como geomorfologia, estudos climáticos, dinâmica populacional e impacto ambiental, requer que os SIGs tenham capacidade de representar os tipos de processos dinâmicos encontrados em estudos de sistemas físicos e sócio-econômicos (DRUCK *et al.*, 2004).

Neste contexto, a Modelagem Dinâmica (BURROUGH, 1998) procura transcender as limitações atuais da tecnologia de Geoprocessamento, fortemente baseada numa visão estática, bidimensional do mundo. O objetivo dos modelos dinâmicos em GIS é realizar a simulação numérica de processos dependentes do tempo, como nos modelos hidrológicos, que simulam o fluxo e transporte de água.

Na definição de Burrough, “um modelo espacial dinâmico é uma representação matemática de um processo do mundo real em que uma localização na superfície terrestre muda em resposta a variações em suas forças direcionadoras”.

Os dados geográficos manipulados pelos SIGs são representados computacionalmente segundo uma arquitetura. A arquitetura mais utilizada é o modelo geo-relacional, também conhecida como arquitetura dual; um sistema gerenciador de bancos de dados (SGBD) relacional armazena em tabelas os atributos dos objetos geográficos, e arquivos gráficos separados para guardar as representações geométricas destes objetos.

Atualmente, com o advento das extensões espaciais incorporadas aos SGBD relacionais, tem-se uma arquitetura integrada, onde os dados cadastrais e espaciais estão armazenados numa única tabela.

2.2 O PROCESSO DE ANÁLISE ESPACIAL

A informação espacial, conjunto de dados manipulados pelos SIGs e que possuem significado além de sua representação própria, é caracterizada em uma das três classes a seguir, quanto ao processo de análise espacial:

- Eventos ou padrões pontuais: fenômenos expressos através de ocorrências identificadas como pontos localizados no espaço, denominados processos pontuais;
- Superfícies contínuas: estimadas a partir de um conjunto de amostras de campo, regular ou irregularmente distribuídas;
- Áreas com contagens e taxas agregadas: dados de levantamentos populacionais, como os censos, e que se referem a indivíduos localizados em pontos específicos do espaço. Esses dados são agregados em unidades de análise, por questões de privacidade e segurança.

A análise desses dados é composta por um conjunto de procedimentos encadeados que almejam a escolha de um modelo inferencial que considere explicitamente o relacionamento espacial presente no fenômeno. O processo inicia-

se com as ferramentas de análise exploratória e visualização de dados, que permitem perceber tendências e pontos discrepantes. A partir daí é possível estabelecer hipóteses sobre as observações e, então, escolher o modelo inferencial melhor suportado pelos dados (DRUCK *et al.*, 2004).

Os modelos inferenciais espaciais são classificados numa das três categorias: variação contínua, variação discreta e os processos pontuais. A resolução de um problema espacial pode envolver a utilização de um ou mais desses modelos. Cabe ao pesquisador escolher, não o modelo certo, mas o modelo que melhor se ajuste aos dados e que tenha o maior potencial de contribuir para a compreensão do fenômeno em estudo.

2.3 CONCEITOS BÁSICOS EM ANÁLISE ESPACIAL

Alguns conceitos permeiam o processo de análise espacial; entre eles pode-se citar: dependência espacial, autocorrelação espacial, inferência estatística, estacionariedade e isotropia.

A dependência espacial é um conceito chave no entendimento e análise dos fenômenos espaciais. Oriundo da primeira lei da Geografia, enunciada por Waldo Tobler: *“todas as coisas são parecidas, mas coisas mais próximas se parecem mais que coisas mais distantes”*. Ou seja, a maior parte das ocorrências naturais, ou sociais, apresentam entre si uma relação que depende da distância.

A autocorrelação espacial é a representação computacional do conceito de dependência espacial. Este conceito é derivado do conceito de correlação estatística. “Auto” no sentido de que ela decorre de uma variável para com ela mesma, quando medida em diferentes posições no espaço. As medidas de autocorrelação espacial baseiam-se numa mesma idéia: verificar como varia a dependência espacial, a partir da comparação entre os valores de uma amostra e de seus vizinhos.

Como consequência da dependência espacial, a inferência estatística espacial não é tão eficiente quanto no caso de amostras independentes do mesmo tamanho. Ou seja, a dependência espacial leva a uma perda do poder explicativo.

Os principais conceitos estatísticos que definem a estrutura espacial dos

dados relacionam-se aos efeitos de 1^a e 2^a ordem.

O efeito de 1^a ordem está relacionado com o valor esperado, ou seja, com a média do processo no espaço. O efeito de 2^a ordem é a covariância entre duas áreas s_i e s_j . Assim, um processo é chamado estacionário se ambos os efeitos são constantes em toda a região estudada. E é isotrópico se, além de estacionário, a covariância depender somente da distância entre os pontos e não da direção entre eles.

3 MODELAGEM ESPACIAL

3.1 CONCEITO DE MODELO

O conceito de modelos é muito usado nas diversas áreas de aplicação. Por isso é importante defini-lo para melhor entendimento.

De modo geral a palavra modelo pode ser entendida como uma representação abstrata da realidade, estruturada de tal forma que permita compreender o funcionamento total ou parcial dessa realidade ou fenômeno.

Num plano também genérico, modelo, para CRAMER (1971), é uma representação abstrata da realidade, que mostra apenas o que é relevante para uma questão específica, “negligenciando” todos os demais aspectos.

Já MANLIVAUD (1967) define modelo num sentido mais restrito. É uma representação formal de idéias ou conhecimentos acerca de um fenômeno. Essas idéias – chamadas teorias – expressam-se por um conjunto de hipóteses sobre os elementos essenciais do fenômeno e das leis que o regem, as quais geralmente se traduzem sobre a forma de um sistema de equações matemáticas.

Tal racionalização de modelos permite a investigação das conseqüências lógicas das hipóteses, consideradas através de sua comparação com os resultados da experiência. Dessa forma, conhece-se melhor a realidade e pode-se, em conseqüência, atuar com mais eficácia sobre ela.

Em síntese, a palavra modelo refere-se a um conjunto de hipóteses estabelecidas a priori sobre o comportamento de um fenômeno, com base numa teoria já existente ou a partir de novas proposições teóricas.

Nos SIGs a aplicação de modelos visa, segundo LAMBIN (1994), responder às questões clássicas: “Por que?”, “Quando?” e “Onde?”. Um modelo que responda a estas questões é capaz de descrever quantitativamente um fenômeno e prever sua evolução, integrando suas escalas temporal e espacial.

3.2 MODELAGEM – ASPECTOS CONCEITUAIS

Os SIGs atuais trabalham bem com dados localizados na superfície da

terra e estáticos no tempo. Como dito na seção 2.1 há um movimento em direção à extensão dos SIGs para que estes sejam capazes de manipular as variações espaço-temporais.

Tipicamente, SIGs são desenvolvidos a partir de suposições pré-estabelecidas quanto a homogeneidade, uniformidade e universalidade das propriedades de seus principais componentes, que incluem o espaço e as relações espaciais, o tempo e o modelo matemático que descreve o fenômeno. Entretanto, para modelar processos dinâmicos em SIGs é necessário flexibilizar estas suposições rígidas, de tal forma que seja possível representar os fenômenos não estáticos no tempo, como por exemplo a própria variação irregular de um processo no tempo (COUCLELIS, 1997).

3.2.1 O Espaço e o Tempo

O espaço é o conceito chave na geografia e, por extensão, na Ciência da Informação Espacial. Tradicionalmente, os geógrafos fazem uma distinção entre os conceitos de espaço absoluto e espaço relativo.

“Espaço absoluto, também chamado Cartesiano ou Newtoniano, é um *container* de coisas e eventos, uma estrutura para localizar pontos, trajetórias e objetos. Espaço relativo, ou Leibnitziano, é o espaço constituído pelas relações espaciais entre coisas e eventos” (COUCLELIS, 1997).

SANTOS (1996) refere-se à distinção entre espaço absoluto e espaço relativo como o “espaço dos fixos” e o “espaço dos fluxos”. Em termos de representações computacionais pode-se, de forma aproximada, traduzir estes conceitos como a distinção entre as representações associadas a recobrimentos planares (mapas de polígonos e matrizes) e representações associadas à conectividade (grafos).

Um caso típico de medida realizada no espaço absoluto são os índices de auto-correlação espacial. Neste caso, um dos instrumentos básicos é a matriz de proximidade espacial, cujo cálculo usualmente é feito em função de distância euclidiana entre objetos ou da existência de uma fronteira entre eles.

Um exemplo de fenômeno em que a dimensão espacial requer o conceito

de espaço relativo é o caso de fluxo de pessoas pela rede de transporte metroviário de uma cidade.

COUCLELIS (1997) propõe a idéia de espaço próximo como uma extensão dos conceitos de espaço absoluto e relativo. No espaço próximo o conceito chave é a vizinhança associada à noção de proximidade, que conduzem também ao conceito de proximidade funcional ou influência. O conceito de vizinhança é facilmente visualizado em representações matriciais do espaço. Algumas operações espaciais disponíveis em SIGs como filtros espaciais, por exemplo, utilizam a noção de espaço próximo de forma limitada. No filtro espacial, o estado de uma célula (um pixel de uma imagem) é modificado com base nos estados das demais células em sua vizinhança, definida através de uma máscara.

Com relação ao tempo, este pode ser representado em diferentes estruturas, com base em três aspectos da representação temporal: granularidade, variação e ordem no tempo.

A ordem temporal refere-se ao modo como o tempo flui. Neste caso, pode-se assumir que o tempo flui de forma linear, ramificada ou cíclica. No tempo linear considera-se que o tempo flui seqüencialmente, ou seja, existe uma ordem de precedência entre os pontos no tempo, de tal forma que cada ponto tenha apenas um sucessor e um antecessor. No tempo ramificado múltiplos pontos podem ser os sucessores ou antecessores imediatos de um mesmo ponto. O tempo cíclico é utilizado para modelar eventos e processos recorrentes (EDELWEISS e OLIVEIRA, 1994).

Com relação à variação temporal duas possibilidades podem ser consideradas: tempo contínuo e discreto. Uma variável temporal contínua é usada em processos que demandam medidas de tempo com níveis arbitrários de precisão. Por exemplo, a expansão da área de desmatamento de uma floresta entre dois instantes (t e t') pode ser interpolada.

Uma variável temporal discreta é usada quando o tempo é medido em certos pontos ou intervalos e a variação é descontínua entre estes pontos. Uma delimitação de lotes de um cadastro imobiliário pode ocupar uma posição num instante t e outra num instante t' , mas não faz sentido dizer que a delimitação ocupou alguma posição intermediária entre t e t' .

Associado ao conceito de variação temporal discreta, existe o conceito de *chronon*. Um *chronon* é a menor duração de tempo suportada por um sistema e pode variar em diferentes aplicações (EDELWEISS e OLIVEIRA, 1994).

A granularidade temporal de um sistema está diretamente relacionada com a duração de um *chronon*. As diferentes granularidades de um sistema temporal conduzem à definição de instante e intervalo de tempo. Um instante de tempo representa um ponto particular no tempo, um intervalo é o tempo decorrido entre dois instantes e um período consiste de uma seqüência de intervalos de tempo .

A incorporação da dimensão temporal em um sistema de informação não se restringe apenas à questão da representação do tempo, mas inclui também questões relativas a sua recuperação. Um SIG temporal também deve ser capaz de recuperar informações através de consultas definidas sobre critérios temporais.

3.3 TIPOS DE MODELOS

Um modelo é constituído de pelo menos três elementos: variáveis, relacionamentos e processos. Ao se construir um modelo, dependendo do objetivo, pode-se dar ênfase a um ou outro destes elementos. Nesta visão, os modelos podem ser classificados em empíricos e de sistemas. Modelos empíricos focalizam os relacionamentos entre as variáveis do modelo, a partir da suposição de que os relacionamentos observados no passado continuarão no futuro (PEDROSA, 2003).

Modelos de sistemas são descrições matemáticas de processos complexos que interagem entre si, enfatizando as interações entre todos os componentes de um sistema (LAMBIN *apud* PEDROSA, 2003).

Uma taxonomia para classificação dos diferentes tipos de modelos é apresentada na figura 3.1.

Os modelos empíricos possuem três componentes chaves: uma configuração inicial, uma função de mudança e uma configuração de saída.

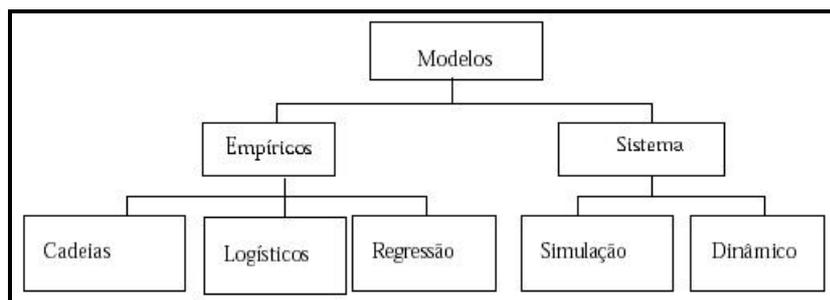


FIGURA 3.1 – Tipos de modelos
 Fonte: LAMBIN *apud* PEDROSA (2003)

A configuração inicial de um modelo dinâmico pode ser obtida através de dados históricos do fenômeno em estudo, chamados de séries temporais. Neste caso, equações diferenciais (totais ou parciais) que incluem pelo menos um termo derivado no tempo podem ser utilizadas para representar o modelo e o processo é classificado como determinístico. Quando variáveis aleatórias são utilizadas para explicar um sistema o processo é classificado como estocástico-probabilístico (PEDROSA, 2003).

Os modelos empíricos são caracterizados pela simplicidade dos modelos matemáticos empregados e pelo número reduzido de variáveis envolvidas. Este modelos são eficientes em fazer predições, embora apresentem limitações em abordar a evolução espacial e identificar os aspectos causais do sistema. São exemplos de modelos empíricos as cadeias de Markov, os modelos logísticos de difusão e os modelos de regressão.

Os modelos de sistemas procuram descrever o sistema como um todo, isto é, tentam representar as interações entre todos os seus componentes. Uma característica chave destes modelos é a eficiência com que abordam a dimensão espacial, implementando conceitos como as relações de vizinhança e suportando o uso combinado de múltiplas escalas. Duas classes de modelos de sistema podem ser citados: os modelos de simulação de ecossistemas e os modelos de simulação dinâmica espacial.

4 ALGORITMOS GENÉTICOS (AGs)

Nesta seção será apresentada uma breve revisão bibliográfica sobre os AGs. Inicia-se a revisão com o histórico e com a apresentação da estrutura básica destes algoritmos. Posteriormente serão apresentados os principais operadores genéticos e a influência dos parâmetros genéticos sobre o desempenho dos mesmos. Ao final desta seção será apresentada uma tendência observada com as heurísticas utilizadas em otimização: a hibridização. No caso a hibridização dos AGs com *Simulated Annealing*.

4.1 INTRODUÇÃO AOS AGs

Um AGs é um tipo de algoritmo de busca que se utiliza do paradigma genético/evolucionário (HOLLAND, 1975). Os AGs foram criados com o intuito de imitar alguns dos processos observados na evolução natural das espécies. Os mecanismos que realizam esta evolução ainda não estão completamente compreendidos, mas algumas de suas características já são bem compreendidas e aceitas. A evolução acontece nos cromossomos, que são os elementos orgânicos responsáveis pela codificação genética dos seres vivos (DAVIS, 1996). As características e fenômenos específicos desta codificação ainda são objetos de muitas pesquisas. Segundo DAVIS (1996), as principais características gerais da teoria evolucionária que já são amplamente aceitas são:

- a) a seleção natural é um processo que atua sobre os cromossomos e, portanto, sobre os seres vivos que eles codificam;
- b) a seleção natural é o elo entre cromossomos e a performance das suas estruturas decodificadas. O processo de seleção natural faz com que os cromossomos que codificam estruturas bem sucedidas se reproduzam mais vezes e com maior probabilidade que as estruturas mal sucedidas;
- c) o processo de reprodução é o ponto onde a evolução acontece. Mutações podem provocar mudanças nos cromossomos dos filhos, fazendo com que eles sejam diferentes dos padrões genéticos dos seus pais, e processos de recombinação podem criar diferentes cromossomos para os filhos, pela combinação dos cromossomos dos pais;

d) a evolução biológica não tem memória. Tudo o que se sabe sobre como produzir indivíduos bem adaptados ao seu meio ambiente está contido no seu genoma - o conjunto de cromossomos carregados pelos indivíduos da população atual - e na estrutura dos cromossomos decodificados.

No começo dos anos 70, John Holland, quando pesquisava as características da evolução natural, acreditava que, se estas características fossem adequadamente incorporadas a algoritmos computacionais, poder-se-ia produzir uma técnica para solucionar problemas difíceis da mesma forma que a natureza fazia para resolver os seus problemas, ou seja, usando a evolução. Acreditando nisto ele deu início a uma pesquisa sobre algoritmos que manipulavam *strings* de 0 e 1, a qual ele deu o nome de cromossomos. Os algoritmos de Holland realizavam a evolução simulada de populações destes cromossomos. Desta forma, imitando a natureza, seus algoritmos resolviam muito bem o problema de encontrar bons cromossomos, através da manipulação do material contido nos cromossomos.

Outro ponto interessante nas técnicas desenvolvidas por Holland é que, assim como na natureza, estes cromossomos não têm conhecimento algum sobre o tipo de problema que estão resolvendo. A única informação que eles dispunham era uma avaliação de cada cromossomo produzido. O objetivo desta avaliação era verificar quais os cromossomos que estavam mais adaptados e, com base nisto, aumentar as suas chances de serem selecionados para a reprodução.

Quando Holland começou os seus estudos sobre estes algoritmos, eles ainda não tinham um nome. Foi apenas quando esta técnica começou a demonstrar o seu potencial que houve a necessidade de se dar um nome adequado e significativo a ela. Como uma referência às suas origens na biologia, Holland os batizou de AGs. De maneira geral, um AG pode ser brevemente descrito através do fluxograma apresentado na figura 4.1.

A técnica usada para codificar as soluções varia de problema para problema e de AG para AG. A codificação clássica usada no trabalho de Holland, e até hoje a mais usada, consistia em usar *strings* de bits, mas com o passar do tempo outros pesquisadores apresentaram outras formas de codificação.

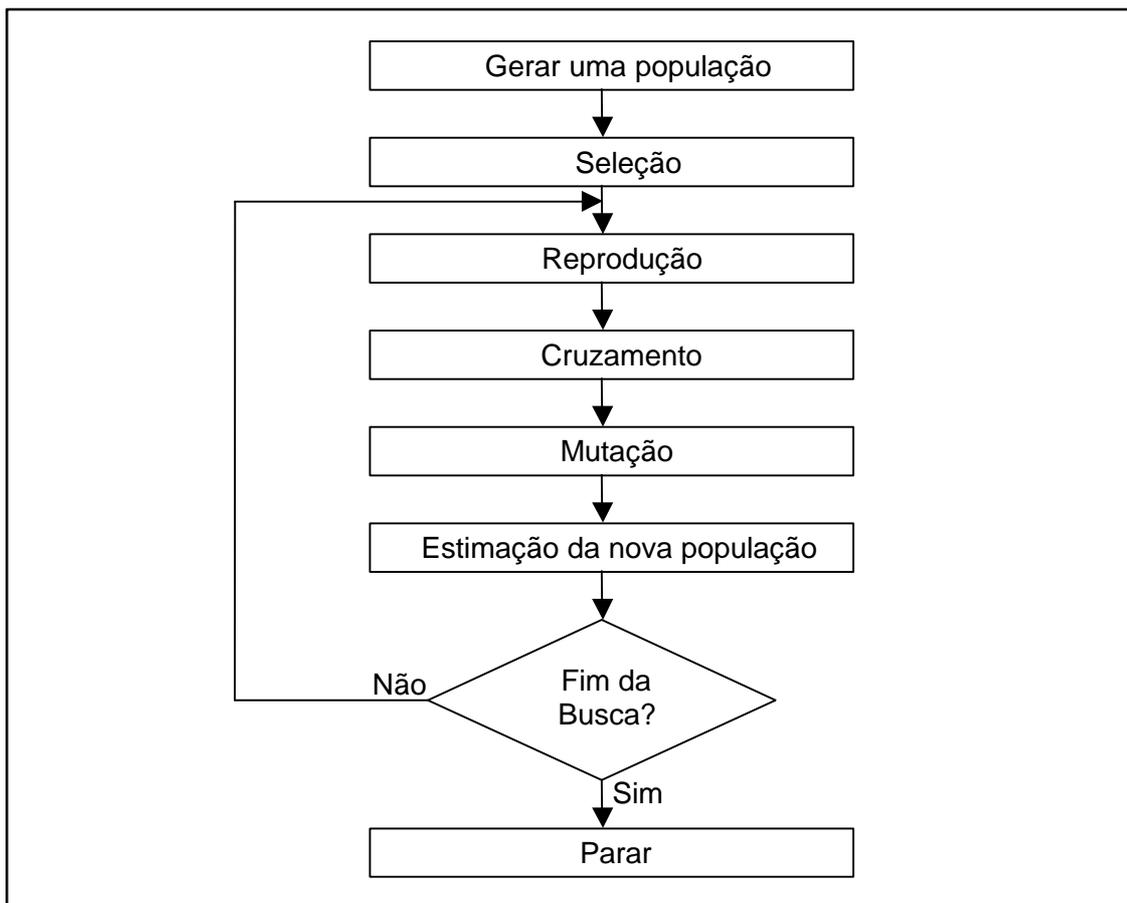


FIGURA 4.1 – Fluxograma que descreve brevemente um algoritmo genético.
 Fonte: CORTES (1999)

A codificação clássica, quando utilizada em problemas que possuem variáveis contínuas e cujas soluções requeridas necessitam boa precisão numérica, torna os cromossomos longos. Para cada ponto decimal acrescentado na precisão, é necessário adicionar 3,3 bits na *string*. (GALVÃO e VALENÇA, 1999)

A consequência imediata do aumento da *string*, que representa o cromossomo, é o aumento no tempo necessário para calcular o equivalente decimal deste cromossomo.

Por este motivo, formas não clássicas de codificação dos cromossomos foram desenvolvidas, gerando codificações adequadas para problemas específicos. (HERRERA, LOZANO e VERDEGAY, 1996)

Uma das formas não clássicas de codificação mais utilizada é a codificação real. Esta forma de codificação consiste em representar, num gene ou cromossomo, uma variável numérica contínua através de seu próprio valor real. Um cromossomo pode ser composto por múltiplos genes quando o problema a ser resolvido envolve duas ou mais variáveis.

As primeiras aplicações da codificação real foram propostas por LUCASIUŠ e KATEMAN (1989) e DAVIS (1989). A partir de então a codificação real tornou-se padrão em problemas de otimização numérica com variáveis contínuas.

CASTRO (1999) afirma que, com certeza, nenhuma forma de codificação funcionaria igualmente bem em todas as situações e que, para cada caso, deve-se fazer uma escolha cuidadosa do tipo de codificação a ser utilizada, pois uma codificação ruim pode não levar ao resultado esperado.

O elemento de ligação entre o AG e o problema a ser resolvido é a função de avaliação. A função de avaliação, chamada de função *fitness*, toma como entrada um cromossomo e retorna um número, ou lista de números, que representam a medida de performance do cromossomo com relação ao problema a ser resolvido. Esta função desempenha no AG o mesmo papel desempenhado pelo meio ambiente na teoria da evolução natural das espécies.

Segundo GOLDBARG e LUNA (2000), os AGs possuem as seguintes características gerais:

- a) Operam em um conjunto de pontos, denominados população, e não a partir de pontos isolados;
- b) Trabalham com um conjunto de parâmetros codificados e não com os próprios parâmetros;
- c) Necessitam como informação somente o valor de uma função objetivo, denominada função de adaptabilidade ou *fitness*;
- d) Usam transições probabilísticas e não regras determinísticas.

4.2 OS OPERADORES GENÉTICOS

HOLLAND (1975) define três técnicas para criar filhos diferentes dos pais: cruzamento, mutação e inversão. Estes três elementos estão intimamente relacionados no modelo básico de um algoritmo genético; os três fazem a evolução da população acontecer.

A finalidade da seleção em um algoritmo é escolher os elementos da população que devem se reproduzir. Em problemas de maximização, esta escolha

deve ser feita de tal forma que dê maior chance de reprodução aos membros da população mais adaptados ao meio ambiente, isto é, àqueles que apresentam um valor da função *fitness* mais elevado. A mais conhecida e utilizada forma de fazer a seleção é a roleta, ou algoritmo Monte Carlo (DAVIS, 1996; MENDES F^o, 2004). Na seqüência apresentaremos o funcionamento da seleção através do mencionado algoritmo.

4.2.1 Seleção por Monte Carlo

Na seleção através do algoritmo Monte Carlo, também conhecida como seleção por roleta, cada indivíduo da população é representado numa roleta proporcionalmente ao seu índice de aptidão. Assim, aos indivíduos com alta aptidão é dada uma porção maior da roleta, enquanto aos de aptidão mais baixa é dada uma porção relativamente menor da roleta. Finalmente, a roleta é girada um determinado número de vezes, dependendo do tamanho da população, e são escolhidos, como indivíduos que participarão da próxima geração, aqueles sorteados na roleta. Um exemplo de aplicação do método da roleta é apresentado na Figura 4.2.

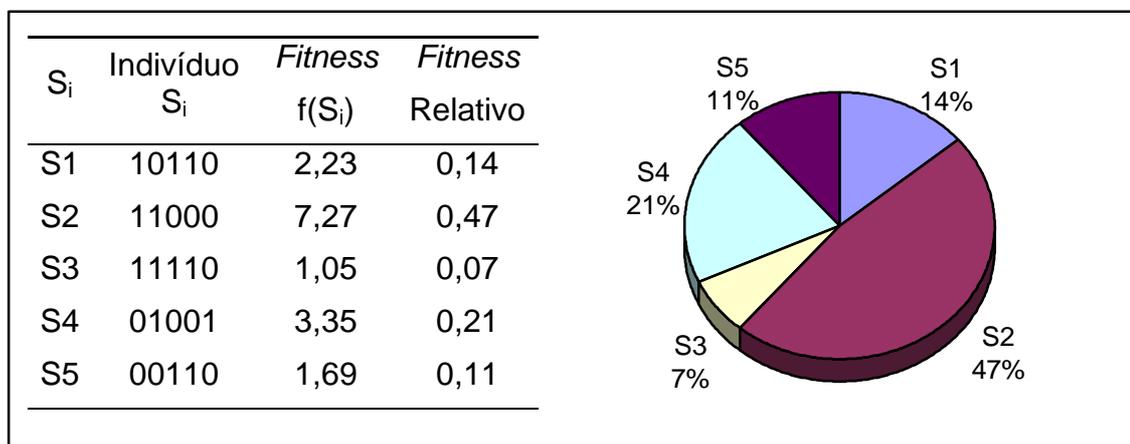


FIGURA 4.2 – Um exemplo de seleção através do algoritmo Monte Carlos ou método da roleta

4.2.2 Elitismo

Para melhorar a convergência dos AGs foi desenvolvida uma técnica chamada elitismo. O elitismo é a técnica mais utilizada para melhorar a convergência destes algoritmos. Ele foi primeiramente introduzido por Kenneth De Jong, em 1975,

e é uma adição aos métodos de seleção que força os AGs a reter um certo número de "melhores" indivíduos em cada geração (YEPES, 2004). Tais indivíduos podem ser perdidos se não forem selecionados para reprodução ou se forem destruídos por cruzamento ou mutação. Em outras palavras, o elitismo seleciona os melhores cromossomos de uma população e transporta-os à geração seguinte. Esta técnica consiste basicamente em realizar o processo de seleção em duas etapas:

- a) Seleciona-se um elite de r membros entre os melhores da população inicial, os quais são incorporados diretamente na população final;
- b) O restante da população final é obtida a partir dos $(n - r)$ elementos restantes da população inicial de tamanho n .

Em geral a elite tem um tamanho reduzido, com $r = 1$ ou 2 para um $n = 50$. Quando é utilizada a técnica do elitismo, o algoritmo converge mais rapidamente. Como na natureza, os indivíduos mais aptos podem, além de reproduzir-se mais, ter uma vida mais longa, muitas vezes sobrevivendo de uma geração para a outra e se reproduzindo. O efeito negativo desta estratégia prende-se ao fato de que a população inicial pode convergir para uma população homogênea de super-indivíduos, não explorando outras soluções.

4.2.3 Cruzamento e Mutação

O objetivo final de ambos é fazer com que os cromossomos criados durante o processo de reprodução sejam diferentes dos cromossomos dos pais. O operador de cruzamento é responsável por combinar os cromossomos dos pais na criação dos cromossomos filhos, e o operador de mutação é responsável pela introdução de pequenas mudanças aleatórias nos cromossomos dos filhos. Vários tipos de operadores de cruzamento foram desenvolvidos por vários pesquisadores, alguns adequados a um tipo específico de codificação dos cromossomos, outros com intenção de serem mais genéricos. Mencionaremos aqui apenas os operadores mais comumente utilizados.

O operador mutação de bit é aplicável em todas as formas binárias de representação de cromossomos. O processo de mutação de bit é bem simples, e

normalmente é realizado da seguinte maneira: dada uma certa probabilidade de mutação, normalmente muito baixa e determinada de forma empírica, cada bit na *string* do cromossomo é avaliado para saber se este bit deverá sofrer uma mutação; caso este bit deva sofrer mutação, o seu valor é simplesmente trocado por um valor determinado aleatoriamente entre os valores que podem ser assumidos pelo cromossomo. A tabela 4.1 mostra 3 cromossomos de comprimento 4 e os números aleatórios gerados para cada um dos bits no cromossomo, os novos bits que demonstram as possibilidades de mutação e o resultado final após a mutação. Os números em negrito na coluna Nos Aleatórios indicam probabilidades muito baixa e, portanto, serão os genes que sofrerão mutação. Os dígitos em negrito na coluna Cromossomo novo são os genes alterados.

TABELA 4.1 – EXEMPLOS DE MUTAÇÃO DE BIT

Cromossomo Anterior	N ^{os} Aleatórios				Novo bit	Cromossomo novo
0011	0,653	0,001	0,287	0,373	1	0111
1001	0,721	0,432	0,043	0,840	-	1001
1110	0,002	0,076	0,934	0,471	0	0110

Quando utiliza-se a codificação em números reais a mutação pode ser realizada de diversas formas: uniforme, gaussiana, *creep*, limite, não-uniforme e não-uniforme múltipla. As três últimas formas de mutação foram propostas por MICHALEWICZ (1994).

A mutação uniforme consiste em substituir o gene selecionado do cromossomo por outro gene gerado aleatoriamente, segundo uma distribuição uniforme, entre os limites mínimo e máximo permitidos. A mutação gaussiana consiste em substituir o gene selecionado por outro gerado a partir de uma distribuição $N(\mu, \sigma^2)$, onde μ é igual ao valor de gene a ser substituído e a variância é definida pelo pesquisador. GALVÃO e VALENÇA (1999) citam que o valor da variância pode ser diminuído à medida que aumenta o número de gerações do algoritmo genético.

A mutação *creep* consiste em acrescentar ou subtrair um pequeno número aleatório obtido de uma distribuição $N(0, \sigma^2)$ onde a variância assume um valor pequeno. Esta mutação é usada para explorar localmente o espaço de busca.

A mutação não-uniforme consiste na simples substituição de um gene por um número extraído de uma distribuição não-uniforme. A mutação não-uniforme múltipla consiste em aplicar a mutação não-uniforme em todos os genes do cromossomo selecionado.

O operador de cruzamento em um ponto é a técnica de cruzamento mais simples e a mais utilizada. Esta técnica consiste em dividir os cromossomos selecionados num ponto de sua cadeia, ponto este escolhido aleatoriamente. Após isso, copia-se para os novos cromossomos uma parte de cada um dos cromossomos selecionados - cromossomos pais, formando assim os novos cromossomos - cromossomos filhos. Nas implementações mais tradicionais, é comum um par de cromossomos selecionados dar origem a dois filhos, mas este não é um fator restritivo. A princípio, pode-se criar qualquer quantidade de filhos, desde que, é claro, o número de alelos permita o número desejado de combinações diferentes. A Figura 4.3 apresenta um exemplo do operador de cruzamento em um ponto.

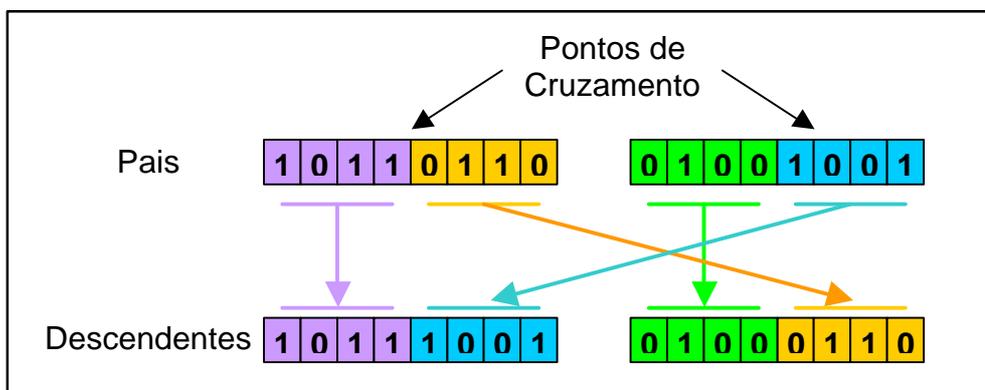


FIGURA 4.3 – Um exemplo do operador de cruzamento em um ponto.
 Fonte: YEPES (2005)

Outra técnica de cruzamento, um pouco menos utilizada que a de cruzamento em um ponto, é o cruzamento em múltiplos pontos. Esta técnica divide o cromossomo em vários pontos e os recombina para formar os filhos, assemelhando-se mais ao processo que ocorre na vida real; possui a vantagem de assegurar uma variedade genética maior.

Nos AGs com codificação real estes operadores de cruzamento não são adequados, pois apenas trocam os valores dos genes, não criando novos valores. Assim, os operadores de cruzamento aritméticos são mais indicados. Alguns operadores de cruzamento aritméticos são: média (DAVIS, 1996), média geométrica,

BLX- α (ESHELMAN e SHAFFER, 1993), aritmético e heurístico (MICHALEWICZ, 1994).

Os cruzamentos média e média geométrica consistem em gerar um novo cromossomo usando a média simples e a média geométrica de dois cromossomos pais, respectivamente.

O cruzamento BLX- α consiste em gerar um novo cromossomo a partir da seguinte expressão:

$$c = p_1 + \mathbf{b}(p_2 - p_1) \quad (\text{eq. 4.1})$$

onde c é o novo cromossomo gerado, p_1 e p_2 são os cromossomos pais e $\beta \in U(-\alpha, 1 + \alpha)$. α é um pequeno valor que estende os limites para a definição de c . Caso o cromossomo seja formado por múltiplos genes a eq. 4.1 é aplicada a cada par de genes de p_1 e p_2 .

O cruzamento aritmético consiste em gerar dois cromossomos filhos (c_1 e c_2) a partir de dois cromossomos pais (p_1 e p_2), usando a expressão:

$$\begin{aligned} c_1 &= \mathbf{b}p_1 + (1 - \mathbf{b})p_2 \\ c_2 &= (1 - \mathbf{b})p_1 + \mathbf{b}p_2 \end{aligned} \quad (\text{eq. 4.2})$$

onde $\beta \in U(0, 1)$.

O cruzamento heurístico consiste em gerar um cromossomo filho a partir de uma interpolação linear entre os pais usando a informação da aptidão. Dados dois cromossomos p_1 e p_2 em que p_1 é melhor do que p_2 em termos de aptidão. Então é produzido um cromossomo c da seguinte forma:

$$c = p_1 + r(p_1 - p_2), \text{ onde } f(p_1) > f(p_2) \quad (\text{eq. 4.3})$$

onde $r \in U(0, 1)$.

Se compararmos os dois esquemas de reprodução, veremos que no esquema de reprodução sexuada é necessário haver mais de um tipo de indivíduo, estes indivíduos devem ter diferenças significativas em alguns aspectos, e devem desprender uma boa parcela de seu tempo e energia para encontrar um parceiro certo para a reprodução. Isto representa um custo a mais para o indivíduo/ algoritmo. Porém, como o esquema de reprodução sexuada parece ter vencido esta guerra,

pode-se concluir que este talvez seja um preço pequeno a pagar, comparado aos benefícios que ele traz consigo.

Um benefício proporcionado pela reprodução sexuada é a combinação rápida de características benéficas, o que não é possível no caso da reprodução assexuada. Uma das formas de vida que mais demonstra possuir uma alta capacidade de adaptação reproduz-se assexuadamente, o vírus. O alto poder de adaptação dos vírus vem do fato de que eles são altamente mutáveis, o que pode nos levar a concluir que a capacidade de sofrer mutações também é uma determinante nos organismos naturais. Ainda que não tenhamos cruzamento, se tivermos uma taxa de mutação bastante elevada, nossa população poderá ser capaz de comportar-se como os vírus, mudando sempre para se adaptar ao seu meio ambiente, e reproduzindo-se de forma assexuada.

4.2.4 O Operador de Inversão

O operador de inversão, assim como a mutação, atua sobre um único cromossomo. Ele inverte a ordem dos elementos entre dois pontos escolhidos aleatoriamente no cromossomo. Apesar deste operador ter sido inspirado por processos naturais, ele gera elevada sobrecarga de trabalho, degradando a performance do algoritmo. Na prática, este operador não é muito utilizado (DAVIS, 1996).

4.3 PARÂMETROS GENÉTICOS

É importante também, analisar de que maneira alguns parâmetros influem no comportamento dos AGs, para que se possa estabelecê-los conforme as necessidades do problema e dos recursos disponíveis.

a) Tamanho da População: O tamanho da população determina o número de cromossomos na população, afetando o desempenho global e a eficiência dos AGs. Com uma população pequena o desempenho pode cair, pois a população fornecerá uma pequena cobertura do espaço de busca do problema. Uma grande população geralmente fornece uma cobertura representativa do domínio do

problema, além de prevenir convergências prematuras para soluções locais ao invés de globais. No entanto, para se trabalhar com grandes populações, são necessários maiores recursos computacionais, ou que o algoritmo trabalhe por um período de tempo muito maior;

- b) Taxa de Cruzamento: Determina a probabilidade com que um cruzamento ocorrerá. Quanto maior for esta taxa, mais rapidamente novas estruturas serão introduzidas na população. Mas se esta for muito alta, a maior parte da população será substituída, e pode ocorrer perda de estruturas de alta aptidão. Com um valor baixo, o algoritmo pode tornar-se muito lento;
- c) Taxa de Mutação: Determina a probabilidade de ocorrência de uma mutação. Uma baixa taxa de mutação previne a convergência prematura para um ótimo local, possibilitando ao algoritmo explorar melhor todo o espaço de busca. Uma taxa de mutação muito alta faz com que o processo de busca torne-se essencialmente aleatório;
- d) Intervalo de Geração: Controla a porcentagem da população que será substituída durante a próxima geração. Com um valor alto, a maior parte da população será substituída, podendo ocorrer perda de estruturas de alta aptidão. Com um valor baixo, o algoritmo pode tornar-se muito lento.

4.4 HIBRIDIZAÇÃO

A técnica de hibridização resulta na integração de uma boa maneira convencional de resolver um problema aos conceitos usuais de AGs. O resultado costuma ser melhor que o obtido com qualquer uma das duas técnicas isoladamente (DAVIS, 1996). A hibridização agrega a representação usual de dados no domínio original, bem como as técnicas de otimização usual já existentes. Isto permite a incorporação de heurísticas otimizadoras ao conjunto de operadores genéticos (recombinação e mutação) que passam portanto a ser dependentes do domínio. Nesse sentido, o algoritmo genéticos passa a ser muito mais uma filosofia de otimização do que um método pronto para utilização.

Um exemplo de hibridização possível é quando o problema exige codificação com base em números reais e não em números binários. Alguns

conceitos teriam que ser adaptados: por exemplo, a mutação não seria mais a troca simples de um *bit*, mas a geração de um novo real, possivelmente dentro de um intervalo dado. Já a recombinação de dois reais poderia ser qualquer número compreendido entre eles, ou talvez a sua média.

Uma heurística utilizada hibridamente com os AGs é o *Simulated Annealing*. A seção a seguir apresenta maiores detalhes desta heurística.

4.4.1 *Simulated Annealing* (SA)

Esta heurística é uma metáfora de um processo térmico utilizado para obtenção de estados de baixa energia num sólido. O processo consiste de duas etapas: na primeira a temperatura do sólido é aumentada para um valor máximo no qual ele se funde; na segunda o resfriamento deve ser realizado lentamente até que o material se solidifique. Nesta segunda fase, executada lentamente, os átomos que compõem o material organizam-se numa estrutura uniforme com energia mínima.

O processo de recozimento (*annealing*) pode ser visto como um processo estocástico de determinação da organização dos átomos num sólido que apresente energia mínima. Em altas temperaturas os átomos movem-se livremente, com grande probabilidade de se moverem para posições que incrementarão a energia total do sistema.

Quando a temperatura baixa, os átomos gradualmente movem-se em direção a uma estrutura regular; somente com pequena probabilidade incrementarão suas energias. Esse processo foi simulado em computador, com sucesso, por METROPOLIS *et al.* (1953). O algoritmo utilizado baseava-se em métodos de Monte Carlo e gerava uma seqüência de estados de um sólido da seguinte maneira: dado um estado corrente i do sólido com energia E_i , um estado subsequente era gerado pela aplicação de um mecanismo de perturbação, o qual transformava o estado corrente em um próximo estado por uma pequena distorção, por exemplo, pelo deslocamento de uma única partícula. A energia do próximo estágio passa a ser E_j . Se a diferença de energia fosse menor ou igual a zero, o estado j era aceito como estado corrente. Se a variação fosse maior que zero, o estado j era aceito com uma probabilidade dada por: $\exp(-(E_j - E_i)/(KB * T))$ onde T representa a temperatura atual

do sistema e KB é uma constante física conhecida como constante de Boltzmann. Essa regra de aceite é conhecida como critério de Metropolis e o algoritmo também leva o seu nome.

KIRKPATRICK *et al.* (1983) desenvolveram um algoritmo de utilização genérica análogo ao de Metropolis, denominado Algoritmo *Simulated Annealing* (Algoritmo de Recozimento Simulado). Nesse algoritmo, utilizaram como critério de aceite uma nova solução, a função:

$$P_{c_k}(\text{aceitar } j) = \begin{cases} 1 & \text{se } g_j \leq g_i \\ \exp\left(\frac{-(g_j - g_i)}{c_k}\right) & \text{se } g_j > g_i \end{cases} \quad (\text{eq. 4.4})$$

onde g é a função a ser otimizada (no caso minimizada), i e j são a solução corrente e uma solução candidata, respectivamente e c_k um parâmetro representando a temperatura T .

Segundo a eq. 4.4, se uma solução candidata j é melhor que a solução corrente i , ou seja ($g_j \leq g_i$), esta é aceita com probabilidade 1. Caso contrário a solução candidata é aceita com uma dada probabilidade. Essa probabilidade é maior à medida em que for menor a variação de energia, definida por ($g_j - g_i$). Ao mesmo tempo, à medida que há um decréscimo da temperatura c_k , o algoritmo torna-se mais seletivo, passando a aceitar, com menor freqüência, soluções que apresentem grande aumento na variação de energia, isto é, soluções que sejam muito piores que a solução corrente. Essa probabilidade tende a zero à medida que a temperatura se aproxima do ponto de congelamento.

O algoritmo SA pode ser considerado como uma extensão do método original de busca local. A busca local requer somente a definição de um esquema de vizinhança, e um método de avaliação do custo de uma solução em particular, sempre apresentando uma solução final. Entende-se por esquema de vizinhança o mecanismo apropriado, dependente do problema que está sendo tratado, através do qual se obtém uma nova solução, também pertencente ao espaço de soluções do problema, realizando uma pequena alteração na solução corrente.

O método de busca local é ineficiente quanto à armadilha do ótimo local, fazendo desse método uma heurística pobre para muitos problemas de otimização

combinatorial. Uma propriedade desejável de qualquer algoritmo é a habilidade de encontrar uma boa solução, independente do ponto de partida. Um ótimo local se caracteriza quando o algoritmo atinge uma região correspondente ao fundo de um vale, em se tratando de um problema de minimização, que não contém a solução ótima e dele não consegue sair, uma vez que todas as soluções naquela vizinhança possuem valores maiores que a solução corrente.

Uma estratégia para escapar da armadilha do ótimo local é executar diversas vezes o algoritmo com diferentes soluções iniciais, sendo adotado como solução a melhor solução encontrada. Entretanto, esse procedimento conduz a um novo problema que é o de determinar quando parar o algoritmo, além de poder ser inviável em se tratando de grandes problemas (ARAUJO, 2001).

O algoritmo SA consegue escapar de um ótimo local uma vez que o aceite de uma nova solução não depende única e exclusivamente do seu valor. Mesmo apresentando um valor pior que o da solução corrente, uma nova solução pode ser aceita de forma probabilística.

4.4.2 O Algoritmo SA

Uma característica muito interessante do algoritmo SA é a simplicidade de sua implementação computacional, conforme mostrado na figura 4.4.

```
Ler  $\alpha$  e NR; //Constante e número de repetições
S = S(); //Conjunto aleatório de soluções iniciais
T = LS; //Limite superior
TMIN = LI; //Limite inferior

Enquanto (T > TMIN) faça
  Para i = 1 até NR faça
    Gerar uma solução S' a partir de S; //perturbação de S
    Avaliar a variação de energia; // $\Delta E = g(S') - g(S)$ ;
    Se (variação de energia <= 0) então S = S'
    Senão
      Gerar aleatoriamente Rnd; //no intervalo [0, 1]
      Se (Rnd < exp(-variação / T) então S = S';
    Fim se;
  Fim Para;
  T = T *  $\alpha$ ;
Fim enquanto;
```

FIGURA 4.4 – O algoritmo SA

Para evitar a convergência precoce para um mínimo local, o algoritmo

inicia com um valor de T relativamente alto. Esse parâmetro é gradualmente diminuído e, para cada um dos seus valores, são realizadas várias tentativas (NR) de se alcançar uma melhor solução, nas vizinhanças da solução corrente.

A expressão $T = T * a$ corresponde ao processo de diminuição da temperatura, normalmente o parâmetro a é uma constante menor que um.

4.4.3 Exemplo de Hibridização de AG com SA

Uma hibridização de AG com SA, simples e possível de ser realizada, é descrita a seguir.

Cada solução, de um conjunto de soluções possíveis, é manipulada através do algoritmo SA. Cada instância do algoritmo SA gera então diversas soluções candidatas que são avaliadas pela função objetivo. Apenas a melhor solução encontrada em cada uma das instâncias é armazenada numa tabela de soluções.

Esta tabela de soluções será a população de indivíduos manipulados por um AG. Sobre os elementos desta tabela serão utilizados os operadores de seleção, cruzamento e mutação.

Um novo conjunto de possíveis soluções é então gerado a partir do AG, constituindo-se no conjunto que será novamente processado pelo algoritmo SA. Esse ciclo se repetirá até que o critério de parada seja atingido.

Além da simplicidade desta hibridização, vê-se também outra característica atraente: a possibilidade de uma implementação que se utilize de processamento paralelo.

5 APLICAÇÕES DE AGs NO PROCESSO DE ANÁLISE DE DADOS GEOGRÁFICOS

Neste capítulo serão apresentados os resultados de um levantamento bibliográfico acerca das aplicações de AGs no processo de análise de dados geográficos. Nessa pesquisa bibliográfica 5 classe de aplicações de AGs que, de algum modo estão relacionados com dados geográficos, foram identificadas.

Algumas ferramentas computacionais desenvolvidas para realizar análise de dados geográficos, ditas “inteligentes”, são também apresentadas neste capítulo.

5.1 CLASSES DE APLICAÇÕES DE AGs NO PROCESSO DE ANÁLISE DE DADOS GEOGRÁFICOS

A primeira classe de aplicações diz respeito aos problemas de otimização em redes, que envolvem, por exemplo, problemas de roteamento de veículos, de distribuição e de projetos de auto-estradas. SKRLEC et al. (1996), RONALD e KIRKBY (1998), TANURE e HAMACHER (1998), MAEDA et al. (1999), REBELLO e HAMACHER (1999), JHA, McCALL e SCHONFELD (2001), TAN et al. (2001), SKOK, SKRLEC e KRAJCAR (2002), TAT e TAO (2003) são exemplos de pesquisadores que realizaram estudos que buscavam solucionar problemas desta classe.

A segunda classe de aplicações é conhecida por *map labeling*. Este problema consiste em espalhar sobre um mapa os nomes de cidades, rios, estradas e demais elementos ali representados, de forma que os textos sejam legíveis, sem sobreposição e estejam alocados próximos das respectivas entidades. Alguns trabalhos desta classe foram desenvolvidos por CHRISTENSEN, MARKS e SHIEBER (1995), ÁLMOS e DMIS (2001), YAMAMOTO e LORENA (2003).

A terceira classe de aplicações são os problemas de concepção de planos de uso de solos. Este problema envolve a distribuição de parcelas de solo à atividades de modo a reduzir riscos ambientais maximizando lucros, por exemplo. Alguns trabalhos envolvendo AGs e sistemas de informações geográficas foram desenvolvidos por BENNETT, ARMSTRONG e WADE (1996), RONALD e KIRKBY (1998), MATTHEWS et al. (1999), MATTHEWS et al. (2000) e BJORNSSON, C. e

STRANGE, N. (s. d.).

A quarta classe de aplicações caracteriza-se pelos problemas de localização de facilidades. Exemplos deste tipo de problema são a localização de escolas numa cidade de forma que os estudantes não necessitem realizar longos percursos, posicionamento de estações rádio-base celular visando atender a maior área possível com o menor número de antenas, localização de abastecedouros comunitários em regiões agrícolas, etc. Alguns trabalhos realizados desta classe foram desenvolvidos por LORENA e NARCISO (2001), NARCISO e LORENA (2002) e DI CHIARA et al. (2003).

A quinta classe de aplicações é a construção de modelos de interação espacial e/ou espaço-temporal. Os trabalhos de STOCKWELL (1993), OPENSHAW e OPENSHAW (1997), FISCHER e LEUNG (1998), XIAO, BENNETT e ARMSTRONG (2000) estudam mecanismos baseados em AGs para criar modelos de interação espacial a partir de vários tipos de dados.

5.1.1 Otimização em Redes

Os AGs aplicados em problemas de otimização em redes visam resolver problemas de roteamento de veículos, de posicionamento de centros de distribuição e projetos de auto-estradas.

SKRLEC et al. (1996) propuseram um algoritmo genético para planejar uma rede primária de distribuição de energia. Os autores modificaram um algoritmo genético utilizado para resolução do problema de roteamento de veículos com múltiplas capacidades. Os dados de rede necessários para o processo de otimização foram extraídos de um sistema de informações geográficas. SKOK, SKRLEC e KRAJCAR (2002) realizaram a integração de um algoritmo genético e um sistema de informações geográficas, baseado no AutoCAD, para solucionar este mesmo tipo de problema.

RONALD e KIRKBY (1998) desenvolveram um algoritmo genético para resolver, simultaneamente, um problema de transporte e de roteamento de veículos. Usaram um esquema de codificação multi-cromossômica, onde cada cromossomo é composto por duas informações. Uma delas referente à rota a ser seguida e outra

refere-se ao veículo a ser utilizado em cada trecho da rota. Propõem ainda, os autores, integrar o algoritmo genético desenvolvido com um sistema de informações geográficas que fornecerá os dados espaciais necessários e uma interface de visualização com o usuário.

TANURE e HAMACHER (1998) propõem o uso de um algoritmo genético para solucionar um problema de distribuição dos correios para a zona sul da cidade do Rio de Janeiro. MAEDA et al. (1999) e TAN et al. (2000) implementaram um algoritmo similar, considerando que os veículos tem uma janela de tempo para atender cada um dos centros de distribuição na rota. Em ambos os trabalhos utiliza-se um sistema de informações geográficas para armazenar a rede sobre a qual o processo de otimização é efetuado.

REBELLO e HAMACHER (1999) implementam o algoritmo genético para solucionar o problema de distribuição dos correios para a zona sul da cidade do Rio de Janeiro. Abordam o problema dividindo-o em dois. Na primeira parte utilizam um algoritmo genético para criar zonas, agrupando centros de distribuição em clusters. Na segunda parte do problema utilizam outro algoritmo genético para encontrar as melhores rotas entre os clusters anteriormente definidos.

JHA e McCALL (2001) e TAT e TAO (2003) utilizam um sistema de informações geográficas e um algoritmo genético para otimizar o traçado de rodovias. Esta otimização baseou-se em escolher os pontos do trajeto de forma a minimizar o custo total da obra, considerando informações físicas, naturais e sócio-econômicas das regiões afetadas. Estas informações estão disponíveis no sistema de informações geográficas.

5.1.2 *Map Labeling*

A distribuição da toponímia sobre um mapa, de modo que as informações fiquem legíveis e posicionadas próximas às respectivas entidades, é um problema combinatório. Problemas combinatórios podem ser resolvidos através de um algoritmo genético.

CHRISTENSEN, MARKS e SHIEBER (1995) realizaram um estudo empírico sobre algoritmos para disposição de rótulos em mapas. Compararam os

algoritmos de força bruta, guloso, gradiente discreto descendente, de Hirsch, de Zoraster e *simulated annealing*. Este último mostrou-se mais eficiente nos testes realizados.

ÁLMOS e DMIS (2001) implementaram um algoritmo genético para dispor rótulos sobre mapas e obtiveram bons resultados com uma estratégia de cromossomo com comprimento variável. Este tipo de cromossomo possibilitou bons resultados mesmo em regiões com muitos rótulos.

YAMAMOTO e LORENA (2003) definiram um algoritmo genético construtivo para o problema de rotular mapas. Em comparações realizadas com 10 diferentes algoritmos este proporcionou melhores resultados mostrando ser adequado para o problema de rotular mapas. Entretanto o tempo computacional para executá-lo é 3,5 vezes superior ao método de busca tabu, que proporciona resultados similares.

5.1.3 Plano de Uso de Solos

Problemas ambientais são decorrentes de decisões não coordenadas sobre o uso de solos. Quando os tomadores de decisão não trabalham em conjunto impactos ambientes significativos podem ocorrer.

BENNET, ARMSTRONG e WADE (1996) mostraram como um algoritmo genético pode ser utilizado para construir um elo de ligação entre critérios de decisão e o espaço geográfico, evoluindo-os mutuamente até atingir soluções aceitáveis para problemas ambientais complexos. Agentes inteligentes foram utilizados para auxiliar os tomadores de decisão a considerar vários critérios, aprender com os sucessos e com os erros das soluções geradas. Este conhecimento pode ser usado para auxiliar na avaliação de soluções alternativas e gerar soluções melhores para problemas ambientais complexos.

MATTHEWS et al. (1999) e MATTHEWS et al. (2000) utilizaram AGs para planejar o uso de solos. Consideraram este problema como um problema de alocação espacial. No primeira trabalho otimizou-se o uso dos solos com base num único critério, o econômico. A evolução apresentada no segundo trabalho diz respeito a multi-otimização; neste trabalho além de se considerar o critério

econômico considerou-se também questões como continuidade das áreas.

BJORNSSON e STRANGE (s. d.) aplicam um algoritmo genético para resolver um problema ambiental que ocorre no oeste da Dinamarca: alagar lotes de terras para atividades agrícolas com o menor custo sem com isso causar desvios significativos no teor Fe^{+2} na água, o que alteraria as condições do habitat do salmão. Os autores informam resultados adequados, ou seja, conseguiu-se uma forma mais econômica de alocar as regiões alagáveis sem com isso prejudicar o habitat dos peixes. Informam também que o algoritmo implementado é computacionalmente caro.

5.1.4 Localização de Facilidades

Os problemas de localização de facilidades caracterizam-se por haver a necessidade de distribuir um número mínimo de facilidades de forma a atingir o maior número possível de beneficiários

LORENA e NARCISO (2001) criaram um algoritmo genético construtivo para resolver este problema na alocação de recursos no campo e na cidade. O algoritmo desenvolvido é genérico, podendo ser aplicado a qualquer número de facilidades e de usuários e mostrou-se eficaz para resolver estes problemas, tidos como *NP-Hard*. NARCISO e LORENA (2002) definiram alterações no operador de mutação do algoritmo genético construtivo, tornando-o ainda melhor.

DI CHIARA et al. (2003) desenvolveram um *framework* baseado num sistema de informações geográficas e num algoritmo genético para solucionar o problema de otimizar a potência e a localização de estações rádio-base. A avaliação do *framework* desenvolvido, aplicando-o em estudos de caso, mostraram que o mesmo é eficiente e preciso. Relatam ainda que o mesmo pode ser utilizado no planejamento de redes.

5.1.5 Construção de Modelos de Interação Espacial

STOCKWELL (1993) construiu um sistema automático para modelar o comportamento de uma espécie biológica a partir de bases de dados. Este sistema,

chamado GARP, tem em seu núcleo um Algoritmo Genético que busca um conjunto ótimo de regras que relaciona uma variável dependente, como a presença de uma determinada espécie vegetal numa região, com um conjunto de variáveis independentes. Este sistema tem sido amplamente utilizado para modelar nichos ecológicos de espécies animais e vegetais; alguns exemplos de aplicação do GARP são o monitoramento de espécies ameaçadas de extinção e predição da expansão de espécies invasoras.

OPENSHAW e OPENSHAW (1997) demonstram como utilizar um algoritmo genético para descobrir relações entre variáveis. Estas variáveis podem representar fenômenos naturais que variam no espaço. Perceber como estas variáveis estão relacionadas é um problema complexo pois matematicamente existem muitas formas de combinar as diferentes variáveis. O algoritmo genético neste caso é utilizado para encontrar quais os coeficientes, operações aritméticas e transcendentais adequados para relacionar as variáveis, ou seja, o modelo matemático que relaciona causas e efeitos. Este algoritmo genético é então chamado de criador de modelos (*model breeder*).

FISCHER e LEUNG (1998) apresentam um algoritmo genético com capacidade para maximizar funções com alto grau de complexidade. Este algoritmo é utilizado para ajustar a topologia de uma rede neural treinando-a e aumentando sua velocidade de convergência. Esta rede é, então, utilizada para modelar interação de dados espaciais.

XIAO, BENNETT e ARMSTRONG (2000) utilizaram um estudo de caso em que modelaram a dispersão e a fixação de sementes de uma espécie de cipreste do pântano (*bald cypress*), caracterizando um processo complexo de otimização espaço-temporal.

5.2 FERRAMENTAS “INTELIGENTES” PARA ANÁLISE DE DADOS GEOGRÁFICOS

A aplicação de algoritmos computacionais na análise de dados geográficos possibilitou o desenvolvimento de uma nova área de conhecimento: a geocomputação.

Dentre os algoritmos computacionais utilizados estão aqueles desenvolvidos na área de inteligência artificial como a programação lógica, as redes neurais e os AGs. Estes algoritmos possibilitaram o desenvolvimento de ferramentas “inteligentes” para análise de dados geográficos.

Nesta seção algumas destas ferramentas, e seus princípios, serão apresentadas permitindo uma melhor visualização da relação entre inteligência artificial e a análise de dados geográficos.

5.2.1 *Model Breeders*

Model Breeders são ferramentas para modelagem automática OPENSHAW (1997). Esta ferramenta são capazes de encontrar um modelo matemático que relaciona variáveis independentes à uma variável dependente, seguindo a expressão geral:

$$y = f(x_1, x_2, \dots, x_n) \quad (\text{eq. 5.1})$$

No ferramenta proposta por Openshaw, o mecanismo utilizado para automatizar este processo foi um Algoritmo Genético. Evoluindo um conjunto de soluções iniciais, obtidas aleatoriamente, este algoritmo é capaz de encontrar um modelo matemático que explique o comportamento de uma variável dependente em função de um conjunto de variáveis independentes.

A evolução é um processo que atua diretamente no cromossomo que foi representado da seguinte forma:

- 1 bit – inclusão ou supressão do gene;
- 3 bits – código da variável independente;
- 2 bits – código da parametrização utilizada (nenhuma, *b, **b, +b);
- 3 bits – código da função utilizada (nenhuma, exp, log, i/x, sen, tan, etc.);
- 2 bits – código do operador binário (+, -, *, /).

Esta ferramenta, segundo o referido autor, apresenta como vantagens a simplicidade, a capacidade de produzir modelos simples de compreender e a

eficiência computacional, quando comparada com métodos tradicionais de modelagem. Como desvantagens tem-se uma representação muito simples dos fenômenos observados e o grande consumo de tempo para obtenção de respostas ótimas.

Um *model breeder* alternativo foi implementado por SANTA CATARINA (2005). O diferencial desta implementação diz respeito à forma de codificação utilizada nos cromossomos genéticos. Em sua proposta Openshaw utilizou-se da codificação binária, com as informações codificadas em cadeias de 0's e 1's. Nesta implementação utilizou-se uma codificação híbrida envolvendo cadeias binárias e números reais em base decimal. Assim, um polinômio geral:

$$y = c_1 \cdot x_i^{Exp_1} op_1 c_2 \cdot x_j^{Exp_2} op_2 \cdots c_n \cdot x_n^{Exp_n} \quad (\text{eq. 5.2})$$

onde:

- y : variável independente;
- c_i : coeficiente de cada termo do polinômio;
- x_{ind} : variáveis independentes;
- Exp_i : expoentes das variáveis independentes;
- op_i : operadores que relacionam os termos do polinômio (+, -, x, /)

foi assim representado:

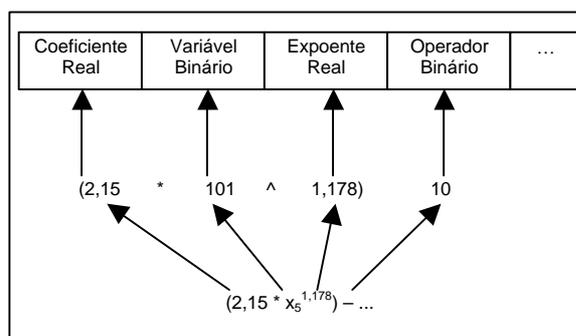


FIGURA 5.1 – Exemplo da codificação empregada no cromossomo

A função de avaliação utilizada baseou-se na soma dos quadrados dos desvios e é calculada pela expressão:

$$Fitness_k = \frac{Min(SQT_1, SQT_2, \dots, SQT_{Tp})}{SQT_k} \quad (\text{eq. 5.3})$$

onde:

- $Fitness_k$: grau de aptidão da k -ésima solução, com $k = 1..Tp$;
- Tp : tamanho da população avaliada;

- *SQT*: somatório dos quadrados dos desvios total:

$$SQT = \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 \quad (\text{eq 5.4})$$

- Y_i : valor assumido pela variável dependente na amostra i ;
- \hat{Y}_i : valor estimado para a variável dependente na amostra i ;
- n : número total de amostras coletadas.

Segundo o autor, a escolha da função (eq. 5.3) foi motivada por sua simplicidade e pela capacidade de medir adequadamente o ajuste do modelo encontrado. Os testes realizados conduziram a resultados considerados bons, permitindo concluir que a ferramenta era adequada para realizar uma análise exploratória dos dados.

5.2.2 *GARP Modelling System – Genetic Algorithm for Rule Set Production*

GARP é um conjunto de módulos desenvolvido para prever a distribuição potencial de entidades biológicas a partir de dados *raster* ambientais e biológicos. Este módulos executam um conjunto diversificado de funções analíticas automaticamente, possibilitando a produção rápida e não-supervisionada de distribuições de animais ou plantas (PAYNE e STOCKWELL, 2001).

O desenvolvimento de mapas de distribuição de espécies impõe uma carga enorme de trabalho num especialista. Cada questão obriga ao especialista a acessar uma base de dados, usando um pacote estatístico de modelagem, preparar e imprimir mapas num GIS. A automação desta tarefa contribui para uma maior disponibilidade de dados diminuindo o tempo de resposta, o custo e ainda liberando especialistas para o desenvolvimento de tarefas mais desafiadoras (STOCKWELL e PETERS, 1999).

GARP é um algoritmo genético que cria modelos de nichos ecológicos para espécies. Os modelos descrevem condições ambientais sobre as quais as espécies podem desenvolver-se. Como dados de entrada, o GARP usa um conjunto de pontos amostrais onde a espécie ocorre e um conjunto de *layers* geográficos que representam os parâmetros ambientais que podem delimitar a sobrevivência da espécie.

A robustez dos AGs é uma característica bem conhecida. O GARP possui uma característica que acentua a capacidade dos AGs de gerar e testar uma ampla faixa de soluções candidatas – a capacidade de gerar e testar diversos tipos de modelos (regras) como modelos categóricos, por faixas e logísticos (STOCKWELL e PETERS, 1999).

Os módulos que constituem o GARP são:

- rasteriz – converte dados pontuais em dados *raster* contínuos (layers);
- presampl – amostra os dados fornecendo conjuntos de dados utilizados no treinamento e nos testes do GARP;
- initial – cria um modelo inicial;
- explain – aplica o algoritmo genético para evoluir o modelo inicial em busca do melhor modelo;
- verify – avalia o modelo produzido pelo módulo anterior utilizando um conjunto independente de dados para teste;
- predict – utiliza o modelo para predizer o valor de cada célula do conjunto de dados *raster*;
- image – converte os dados do módulo predict em imagens;
- translat – traduz para a linguagem natural o modelo encontrado.

A utilização destes módulos segue um processo em lote, na ordem em que foram acima apresentados.

5.2.2.1 Regras

Um algoritmo genético pode ser visto como uma máquina de aprendizado. O algoritmo genético GARP é responsável por criar um conjunto de regras. Cada regra é um modelo em si mesma; um condicional *se-então* utilizado para fazer inferência sobre os valores de uma variável de interesse. O conjunto de regras desenvolvidas pelo GARP são mais precisamente descritas como um modelo inferencial do que como um modelo matemático. Modelos inferenciais diferem de

modelos matemáticos no ponto em que os primeiros estão mais relacionados com a lógica do que com matemática e o processo básico é a inferência lógica ao invés de cálculos (STOCKWELL e PETERS, 1993).

A forma geral de uma regra é visualizada na figura 5.2.

Se A então B e A é verdadeiro, então ocorre B.

FIGURA 5.2 – Forma Geral de uma Regra

A precisão da regra é determinada a partir de cálculos probabilísticos simples. Um conjunto de dados pode estar identificado com a condição de uma regra (por exemplo o conjunto de dados onde a precipitação está entre 600 mm e 700 mm). A probabilidade de ocorrência das espécies pode ser calculada a partir do número de células no qual a espécie ocorre dividido pelo número total de células.

Quatro tipos de regras estão presentes no GARP: regras atômicas, regras BIOCLIM, regras de faixas e regras logísticas.

- Regras Atômicas:

É o tipo mais simples de regra utilizada pelo GARP. Uma regra atômica usa somente um valor para cada variável na condição da regra. A figura 5.3 mostra um exemplo deste tipo de regra.

Se TANN = 23 e GEO = 4 então PRESENT

FIGURA 5.3 – Exemplo de Regra Atômica

- Regras BIOCLIM

Uma regra BIOCLIM está baseada na modelo utilizado no programa BIOCLIM (NIX, 1986). O programa BIOCLIM produz um modelo envelopando os valores ambientais para os quais determinada espécie ocorre; este envelope é definido estatisticamente, tipicamente considerando a faixa do percentil 95. Isto é, o envelope ambiental definido na regra envolve 95% dos pontos de dados onde determinada espécie ocorre. Um ponto analisado é predito como presente se estiver contido dentro do envelope e ausente em caso contrário. A figura 5.4 mostra um exemplo deste tipo de regra.

Se $TANN = (23, 29]$ e $TMNCM = (10, 16]$ e
 $TMXWM = (35, 38]$ e $TSPAN = (19, 72]$ e
 $TCLQ = (21, 23]$ e $TWQM = (23, 30]$ e
 \vdots
 $RCLQ = (1, 16]$ e $RWMQ = (272, 532]$ então $SP = PRESENT$

FIGURA 5.4 – Exemplo de Regra BIOCLIM

Regras BIOCLIM não estão restritas apenas às variáveis climáticas; qualquer variável pode ser usada. Estas regras podem predizer tanto a presença como ausência de uma espécie, mas nunca ambas. A negação de uma regra BIOCLIM pode ser usada para predizer a presença ou ausência de uma espécie.

- Regras de Faixas

É uma generalização das regras BIOCLIM. Numa regra de faixa um número de variáveis podem ser consideradas irrelevantes. Um exemplo deste tipo de regra é apresentado na figura 5.5.

Se $GEO = (6, 244]$ e $TMNEL = (228, 1480]$ então $SP = ABSENT$

FIGURA 5.5 – Exemplo de Regra de Faixas

- Regras Logísticas

Regras logísticas são uma adaptação dos modelos de regressão logísticos. Um regressão logística segue uma equação onde a saída é transformada numa probabilidade. Por exemplo a regressão logística tem como saída uma probabilidade p indicando se uma regra deve ser aplicada. p é calculada por:

$$p = \frac{1}{e^{-y} + 1} \tag{eq. 5.5}$$

onde $y = -(c_0 + c_1 \cdot d_1 + c_2 \cdot d_2 + \dots + c_n \cdot d_n)$ é uma equação obtida por análise de regressão linear múltipla.

Um exemplo deste tipo de regra é apresentado na figura 5.6.

Se $0,1 - GEO * 0,1 + TMNEL * 0,3$ então $SP = ABSENT$

FIGURA 5.6 – Exemplo de Regra Logística

5.2.2.2 Codificação das Regras

Para que as regras sejam manipuladas pelo GARP é necessário que sejam codificadas numa estrutura manipulável computacionalmente. Esta estrutura, nos AGs, recebe o nome de cromossomo. O conjunto de regras apresentados na figura 5.7 foi codificado nos cromossomos apresentados na tabela 5.1.

r_1 : Se $TMIN = (5, 10]$ e $TMED = (10, 22]$ e $ELEV = (1, 2]$ então *PRESENT*
 r_2 : Se $TMIN = (0, 15]$ e $TMED = (0, 50]$ e $ELEV = (0, 20]$ então *ABSENT*
 r_3 : Se $TMIN * 0,80 + TMED * -0,2 + ELEV * 0,45$ então *ABSENT*

FIGURA 5.7 – Conjunto de regras
 Fonte: SIQUEIRA (2005)

TABELA 5.1 – Cromossomos que codificam o Conjunto de Regras da Figura 5.7

Regra	TMIN	TMIN	TMED	TMED	ELEV	ELEV	P/A	f(r)
1	5	10	10	22	1	2	P	50%
2	0	15	0	50	0	20	A	12%
3	0,8	---	-0,2	---	0,45	---	A	95%

5.2.2.3 Mecanismo Evolutivo

O mecanismo evolutivo do GARP utiliza-se de 4 operadores genéticos: a seleção, o cruzamento, a junção e a mutação.

O mais importante operador de recombinação é o cruzamento. Este operador combina partes de dois cromossomos gerando filhos que carregam características dos cromossomos pais. A figura 5.8 mostra um exemplo de aplicação deste operador sobre cromossomos binários.

$x_1 = 100:01010$
 $x_2 = 010:10100$
 \Downarrow
 $y_1 = 100:10100$
 $y_2 = 010:01010$

FIGURA 5.8 – Exemplo de *crossover* sobre cromossomos binários

O cruzamento exerce dois papéis no processo de busca. Primeiro, fornece novos pontos do espaço de busca para serem testados, mantendo um mesmo esquema. Observe que tanto o cromossomo x_1 como y_1 representam o

mesmo esquema 100#####. Segundo, o cruzamento introduz novos esquemas na população. O cromossomo y_2 representa o esquema #1001###, que não está presente em nenhum dos pais. A figura 5.9 mostra um exemplo de aplicação do operador cruzamento no GARP.

Regra 1	5	10	10	22	1	2	P	50%
Regra 2	0	15	0	50	0	20	A	12%
↑								
Ponto de corte								
Regra 4	5	10	10	50	0	20	P	87%
Regra 5	0	15	0	22	1	2	A	9%

FIGURA 5.9 – Exemplo de operação cruzamento sobre as regras no GARP

O operador de junção é ligeiramente distinto do operador de cruzamento; somente um filho é gerado a partir da combinação de dois pais. A figura 5.10 mostra um exemplo do operador de junção

Regra 1	5	10	10	22	1	2	P	50%
Regra 2	0	15	0	50	0	20	A	12%
↑								
Início do corte				Fim do corte				
Regra 6	5	10	0	50	0	2	P	70%

FIGURA 5.10 – Exemplo de operação de junção sobre as regras no GARP

Há dois operadores de mutação disponível no GARP. O primeiro deles, chamado de mutação randômica consiste em substituir um valor qualquer do cromossomo por outro no intervalo entre 1 e 254. O segundo deles, chamada de mutação incremental, consiste em adicionar uma unidade ao valor selecionado no cromossomo. A figura 5.11 mostra um exemplo para cada uma destas mutações.

Regra 1	5	10	10	22	1	2	P	50%
↑								
Regra 7	5	10	10	50	1	2	P	45%
Regra 2	0	15	0	50	0	20	A	12%
↑								
Regra 8	5	10	10	50	1	21	P	15%

FIGURA 5.11 – Exemplo de operações de mutação sobre as regras no GARP

O objetivo do GARP são dois: maximizar a significância e a precisão das regras sem criar o problema de *overfitting*¹ ou regras por demais especializadas. A significância é medida através de um teste χ^2 sobre a diferença entre as probabilidades preditas a priori e a posteriori pela regra. Maximizar a significância e a precisão preditiva é uma inovação nos sistemas analíticos; muitos modelos maximizam apenas a significância.

O processo de avaliação consiste em testar cada uma das regras utilizando o conjunto de dados amostrado pelo programa “presampl”. O valor obtido pela eq. 5.6 será o valor de aptidão da regra.

$$Sig = \frac{pXYs - no \cdot \frac{pYs}{n}}{\sqrt{\frac{no \cdot pYs \cdot \left(1 - \frac{pYs}{n}\right)}{n}}} \quad (\text{eq. 5.6})$$

onde:

- *Sig*: valor de aptidão da regra (significância);
- *pXYs*: número de pontos amostrados que a regra prevê corretamente;
- *no*: número de pontos amostrados avaliados pela regra;
- *pYs*: número de pontos amostrados com a mesma conclusão que a regra;
- *n*: número total de pontos amostrados.

Ordenando-se as regras pelo índice de aptidão inicia-se o processo de seleção. Um limite de corte é estabelecido e, os indivíduos abaixo deste limite, são descartados. Os indivíduos restantes, os mais aptos, passam novamente pelo processo evolutivo, até que o critério de parada seja atingido. A figura 5.12 ilustra o processo de seleção.

Há dois critérios de parada no GARP. O primeiro deles é um número pré-estabelecido de gerações. O segundo consiste em contar o número de melhores regras que são geradas no processo. Caso este número seja inferior a um limiar pré-estabelecido o processo evolutivo pára.

¹ *Overfitting* é um problema que está sempre presente na modelagem. Um modelo que apresenta *overfit* pode ser excelente sobre os dados para o qual foi ajustado, mas ter uma capacidade preditiva muito pobre.

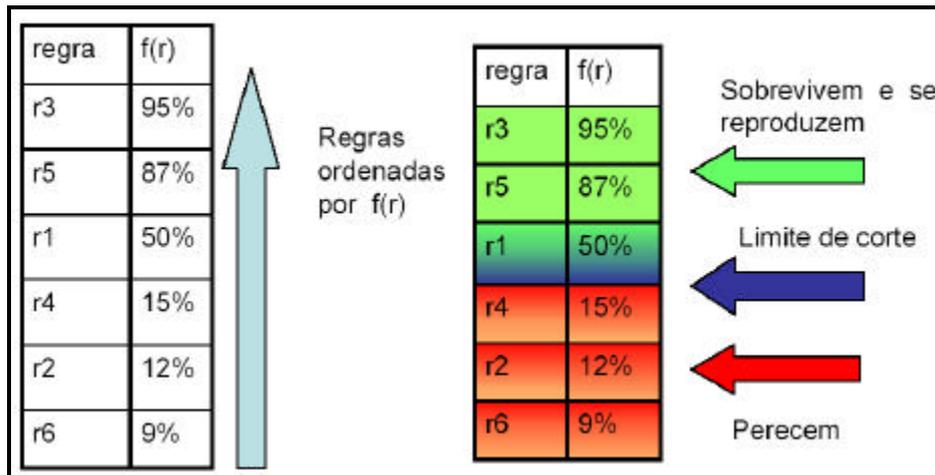


FIGURA 5.12 – O processo de seleção do GARP
Fonte: SIQUEIRA (2005)

5.2.3 DesktopGARP

DesktopGARP é um software para pesquisas em biodiversidade e ecologia que possibilita ao usuário prever e analisar distribuições de espécies selvagens (UNIVERSITY OF KANSAS, 2005).

DesktopGARP é uma implementação do GARP, para computadores padrão IBM-PC, que proporciona uma interface mais amigável ao usuário. A figura 5.13 mostra a interface do sistema.

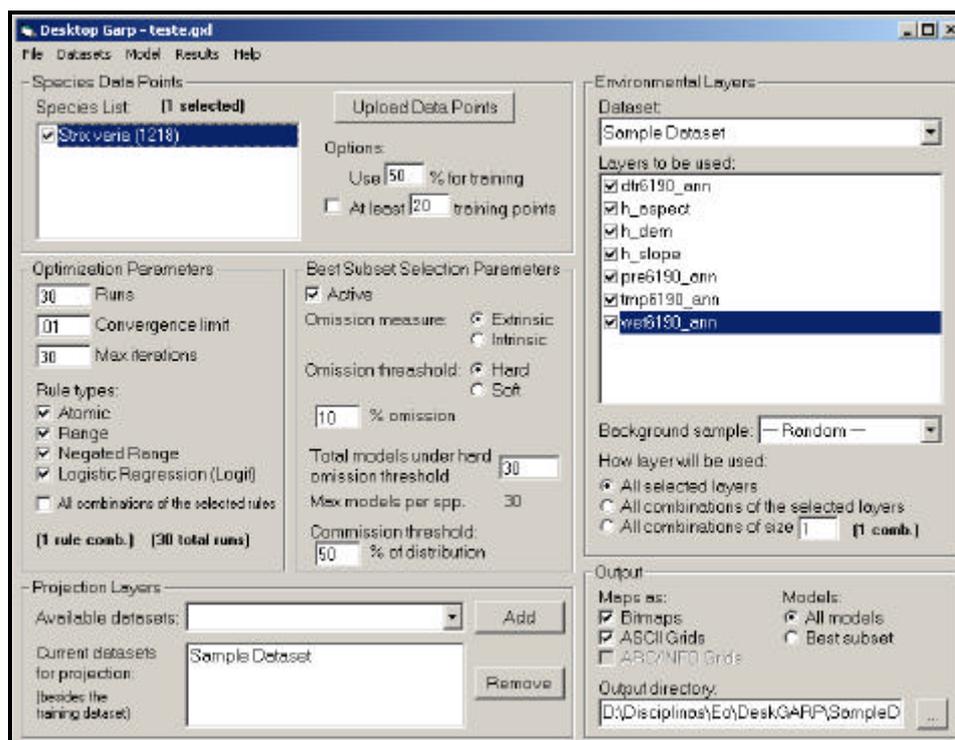


FIGURA 5.13 – Interface do DesktopGARP

Este sistema é uma re-engenharia do sistema GARP. Como inovação, na versão 1.1.6, foi implementado o método de seleção do melhor subconjunto solução. A seção seguinte fornece maiores detalhes sobre este método.

5.2.3.1 Seleção do Melhor Subconjunto Solução

Este método propõe-se a avaliar a qualidade dos modelos ajustados pelo DesktopGARP. Existem, basicamente, duas estratégias para se avaliar a qualidade de um modelo ajustado. A primeira delas é coletar novas amostras e testa-las no modelo. A segunda é dividir as amostras iniciais em dois conjuntos: um conjunto para ajustar o modelo e outro para verificar o modelo.

A partir de um conjunto amostral distinto daquele utilizado para ajustar o modelo é possível construir uma matriz de confusão. Esta matriz é utilizada para quantificar a qualidade do modelo ajustado. A figura 5.14 mostra o formato da matriz de confusão.

	Presente	Ausente
Predição – Presente	A	B
Predição – Ausente	C	D

FIGURA 5.14 – Matriz de Confusão

Os valores A e D são predições corretas. B e C são considerados erros de predição. B é o erro por comissão que gera um falso positivo; C é o erro por omissão que gera um falso negativo.

Os erros do tipo B (comissão) não são considerados erros graves podendo ser causados por diversos fatores:

- A área é adequada à espécie mas não foi amostrada; a espécie pode ser encontrada na área;
- A área é adequada à espécie, mas fatores topológicos e/ou biológicos impedem que a espécie ocupe a área;
- A área é mesmo inadequada – caso de erro verdadeiro.

Os erros do tipo C (omissão) são considerados erros graves. Ou seja, um local onde sabe-se que espécie é encontrada está sendo predito como ausente.

Algoritmos estocásticos como o GARP produzem vários modelos com os mesmos dados de entrada. De posse destes vários modelos é possível calcular seus erros e plotá-los num espaço omissão/comissão. Para uma espécie com um grande número de ocorrências a curva padrão desses erros é apresentada na figura 5.15.

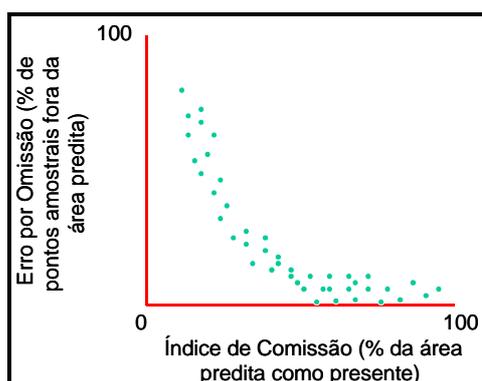


FIGURA 5.15 – Curva padrão para a relação Omissão/Comissão

A figura 5.16 identifica no gráfico as regiões com relação a intensidade dos erros de omissão/comissão.

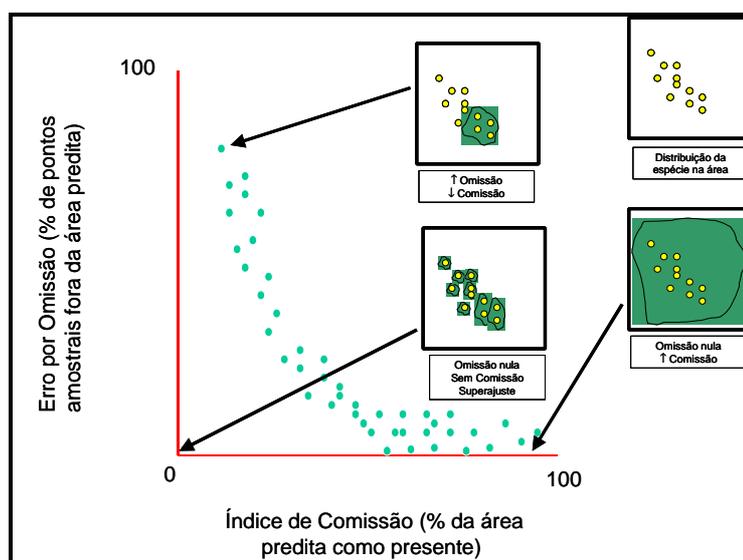


FIGURA 5.16 – Classificação das regiões quanto à intensidade dos erros de omissão/comissão

As regiões com altos erros da classe C (omissão) corresponde a modelos inadequados. A aplicação de um limiar para eliminar esses modelos deve ser então

aplicado. A figura 5.17 mostra como seria a aplicação deste limiar.

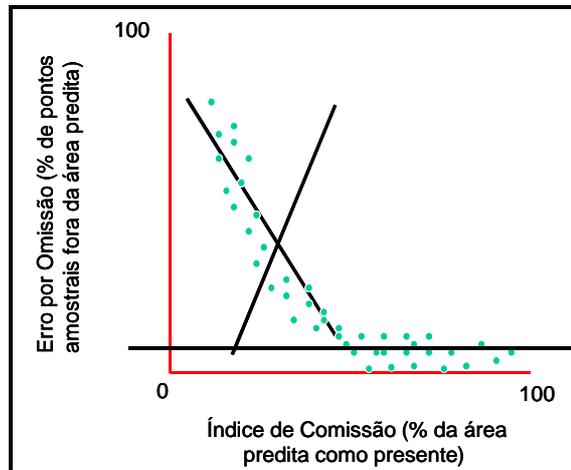


FIGURA 5.17 – Aplicação do limiar para eliminar modelos com muitos erros de omissão

Após a aplicação do limiar ainda restarão os modelos que apresentam superajuste e superpredição. Estes também não são considerados bons modelos. Assim, modelos na região da mediana entre estas classes são considerados os melhores modelos ajustados. Uma representação gráfica desta situação é apresentada na figura 5.18.

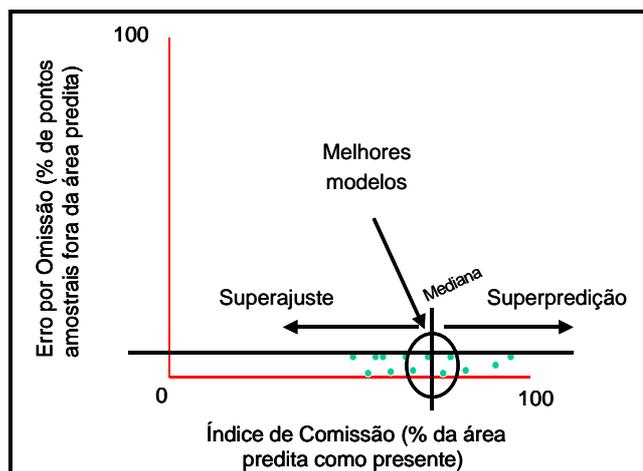


FIGURA 5.18 – Identificação da região com os melhores modelos ajustados

Os parâmetros para a seleção do melhor subconjunto solução são apresentados na figura 5.19 e explicados a seguir.

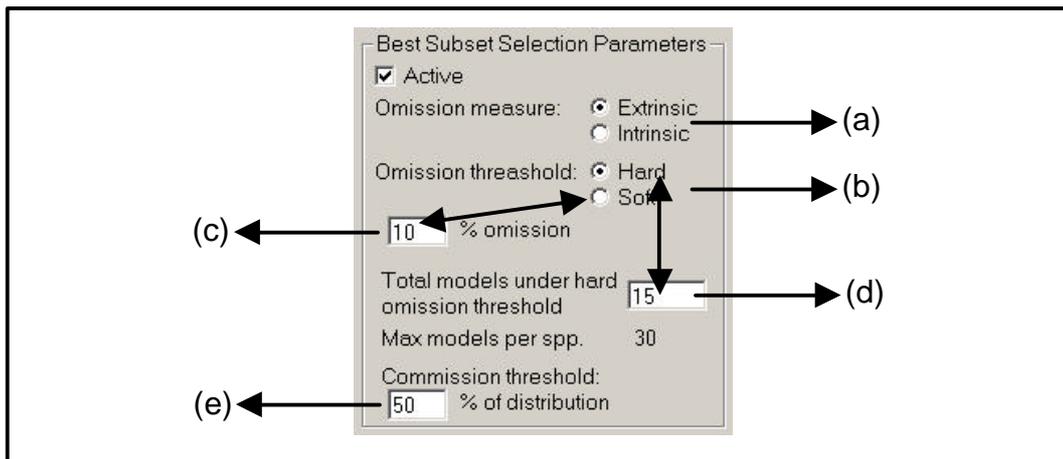


FIGURA 5.19 – Parâmetros para seleção do melhor subconjunto solução

Em (a) seleciona-se como será considerado o erro por omissão. Se o conjunto amostral utilizado para ajustar o modelo for o mesmo que está sendo usado para testá-lo deve-se selecionar a opção “*Intrinsic*”, caso contrário “*Extrinsic*”.

Em (b) define-se como será estabelecido o limiar para eliminar os modelos com altos erros de omissão. A seleção por “*Hard*” permite definir o número (d) de modelos desejados abaixo do limiar. A seleção por “*Soft*” estabelece que o número de modelos desejados abaixo do limiar seja proporcional (c) ao número de modelos construídos.

Já o limiar para os erros de comissão são definidos em (e). A escolha por um valor igual a 50% equivale a dizer que está sendo considerada a mediana dos modelos selecionados abaixo do limiar de omissão.

6 CONSIDERAÇÕES FINAIS

Neste estudo abordou-se as aplicações dos Algoritmos Evolutivos, particularmente os AGs, no processo de análise de dados geográficos. Diversas aplicações foram encontradas na pesquisa bibliográfica realizada como: otimização em redes, planejamento e uso de solos, alocação de facilidades e construção de modelos de interação espacial, por exemplo.

Uma atenção maior foi dispensada aos AGs aplicados no processo de construção de modelos de interação espacial. Três aplicações utilizadas com esta finalidade foram estudadas: *Model Breeders*, GARP e DesktopGARP.

O primeiro deles, o *Model Breeder*, tem como objetivo encontrar o melhor modelo matemático que relacione uma variável dependente com um conjunto de variáveis independentes. O segundo tem como objetivo prever a ocorrência, ou não, de determinada espécie num local do espaço geográfico, dado um conjunto de amostras e um conjunto de layers com informações geográficas e climáticas deste local. O terceiro, o DesktopGARP, é uma evolução do segundo. Trata-se de uma reengenharia do GARP, com os mesmos objetivos iniciais, que visa facilitar a manipulação através de uma interface gráfica amigável.

Nas seções seguintes serão citados alguns erros de código encontrados no GARP, bem como algumas sugestões para melhorias no algoritmo.

6.1 ERROS DE CÓDIGO

O GARP é um sistema de código aberto, disponível no endereço <http://biodi.sdsc.edu/Doc/GARP>. Analisando-se o código fonte do módulo “explain” constatou-se algumas inconsistências, como as apresentadas nas figuras 6.1 a 6.3.

O código apresentado na figura 6.1 apresenta dois erros. O primeiro é a replicação da mutação para um único gene (caso da regra ‘r’ – atômica). O segundo erro deve-se à forma como foi implementada a atribuição dos valores; pode ocorrer que os novos valores venham a ser iguais.

Os códigos apresentados nas figuras 6.2 e 6.3 apresentam o mesmo erro. A condição de teste no laço de repetição indica que o mesmo somente será

executado, uma única vez, se os valores aleatórios xpoint1 e xpoint2 forem iguais.

```
switch (New[i]->Type){
  case 'r':
    New[i]->Gene[j*2] = New[i]->Gene[j*2] +
      Randint(-Temperature, Temperature);
    New[i]->Gene[j*2] = New[i]->Gene[j*2] +
      Randint(-Temperature, Temperature);
    break;
  default:
    New[i]->Gene[j*2] -= Randint(-Temperature, Temperature);
    New[i]->Gene[j*2+1] += Randint(-Temperature, Temperature);
    New[i]->Gene[j*2] = Min(New[i]->Gene[j*2], New[i]->Gene[j*2+1]);
    New[i]->Gene[j*2+1] = Max(New[i]->Gene[j*2], New[i]->Gene[j*2+1]);
}
```

FIGURA 6.1 – Trecho do código da rotina Mutate() – módulo heuristic.c

```
xpoint1 = Randint(0, New[mom]->Length);
xpoint2 = Randint(0, New[mom]->Length);

/* perform crossover */
for(i = xpoint1; i%(New[mom]->Length) == xpoint2; i++){
  temp = kid1[i];
  kid1[i] = kid2[i];
  kid2[i] = temp;
}
```

FIGURA 6.2 – Trecho do código da rotina Crossover() – módulo heuristic.c

```
xpoint1 = Randint(0, New[mom]->Length);
xpoint2 = Randint(0, New[mom]->Length);

/* perform join */
for(i = xpoint1; i%New[mom]->Length == xpoint2; i++){
  temp = kid1[i];
  if (temp != '\255') kid2[i] = temp;
  diff += (kid1[i] != kid2[i]);
}
```

FIGURA 6.3 – Trecho do código da rotina Join() – módulo heuristic.c

Os códigos devidamente corrigidos são apresentados nas figura 6.4 a 6.6.

```
switch (New[i]->Type){
  case 'r':
    New[i]->Gene[j*2] = New[i]->Gene[j*2] +
      Randint(-Temperature, Temperature);
    break;
  default:
    temp1 = New[i]->Gene[j*2] - Randint(-Temperature, Temperature);
    temp2 = New[i]->Gene[j*2+1] + Randint(-Temperature, Temperature);
    New[i]->Gene[j*2] = Min(temp1, temp2);
    New[i]->Gene[j*2+1] = Max(temp1, temp2);
}
```

FIGURA 6.4 – Código corrigido – rotina Mutate() – módulo heuristic.c

```

for (i = xpoint1; i < xpoint2; ++i){
    temp = kid1[i];
    kid1[i] = kid2[i];
    kid2[i] = temp;
}

```

FIGURA 6.5 – Código corrigido – rotina Crossover() – módulo heuristic.c

```

for (i = xpoint1; i < xpoint2; ++i){
    temp = kid1[i];
    if (temp != '\255') kid2[i] = temp;
    diff += (kid1[i] != kid2[i]);
}

```

FIGURA 6.6 – Código corrigido – rotina Join() – módulo heuristic.c

A razão em se analisar o código do módulo “explain” deve-se ao fato de que nele encontra-se implementado o algoritmo genético.

6.2 SUGESTÕES

O algoritmo genético implementado no GARP e no DesktopGARP utiliza uma codificação numérica não binária. O operador de cruzamento trata os cromossomos como uma cadeia de números, escolhendo dois pontos de corte para realizar o cruzamento.

A primeira sugestão é modificar a rotina *Crossover()* para realizar um cruzamento numérico, como os propostos por MICHALEWICZ (1994). A figura 6.7 mostra como é realizado o cruzamento na implementação atual e a figura 6.8 mostra como poderia ser realizado utilizando um cruzamento aritmético por média.

Regra 1	5	10	10	22	1	2	P
Regra 2	0	15	0	50	0	20	A
		↑				↑	
		Ponto de corte			Ponto de corte		
Regra 3	5	15	0	50	0	2	P
Regra 4	0	10	10	22	1	20	A

FIGURA 6.7 – Exemplo de operação cruzamento sobre as regras no GARP

No caso de cruzamento aritmético por média um só filho é gerado. Repetindo o processo novamente, com outros pontos de corte pode-se obter mais filhos. Outra alternativa é utilizar média ponderada, com o fator de ponderação

aleatório.

Regra 1	5	10	10	22	1	2	P
Regra 2	0	15	0	50	0	20	A
		↑				↑	
	Ponto de corte				Ponto de corte		
Regra 3	0	12	5	36	0	20	A

FIGURA 6.8 – Exemplo de cruzamento com operador aritmético por média

A segunda sugestão é a eliminação das regras BIOCLIM, uma vez que as regras por faixas são uma generalização das mesmas. Na mecânica do algoritmo genético implementado não haveria impacto com essa alteração.

A terceira sugestão é a hibridização do AG utilizado no GARP através do algoritmo SA. Neste caso o algoritmo SA seria utilizado para encontrar um bom conjunto de regras que, posteriormente, sofreria o processo de seleção e cruzamento através do AG, gerando assim a nova população que voltaria a ser processada pelo algoritmo SA. Esse ciclo ocorreria até atingir o critério de parada.

Esta hibridização pode levar a uma implementação do GARP que utilize processamento paralelo. O algoritmo SA pode, perfeitamente, ser executado em paralelo; cada instância salvará a melhor solução numa tabela, compondo a população de saída, que será então processada pelo AG.

REFERÊNCIAS BIBLIOGRÁFICAS

1. ÁLMOS, A.; DMIS, A. V. T. Advanced genetic methods for map labeling. **MBE Miniszimpózium helye**. 2001. <<http://www.mit.bme.hu/events/minisy2001/Minisymph2001-Cikkek>>. Visitado em 04/05/2004.
2. ARAUJO, H. A. **Algoritmo simulated annealing: uma nova abordagem**. Florianópolis, 2001. Dissertação (Mestrado em Engenharia da Produção), Universidade Federal de Santa Catarina.
3. BENNETT, D. A.; ARMSTRONG, M. P.; WADE, G. A. Agent mediated consensus-building for environmental problems: a genetic algorithm approach. **Proceedings of Third International Conference on Environmental Modeling and Geographic Information Systems**. 1996. <<http://www.uiowa.edu/~geog/faculty/armstrong>>. Visitado em 10/05/2004.
4. BJORNSSON, C.; STRANGE, N. Heuristic allocation of wetlands in GIS. **Esri Library**. s. d. <<http://gis.esri.com/library/userconf/proc00/professional/papers/PAP238/p238.htm>>. Visitado em 13/05/2004.
5. BURROUGH, P. Dynamic Modelling And Geocomputation. **Geocomputation: A Primer**. LONGLEY, P.; BATTY, M.; McDONNEL, R. London : John Wiley & Sons, 1998.
6. CÂMARA, G.; MONTEIRO, A. M. V. Geocomputation techniques for spatial analysis: are they relevant to health data? **Cadernos Saúde Pública**. v. 17, n. 5, p.1059-1081. Set./Out. 2001.
7. CASTRO, J. P. **Um algoritmo evolucionário para geração de planos de rotas**. Florianópolis, 1999. Dissertação (Mestrado em Engenharia da Produção), Universidade Federal de Santa Catarina.
8. CHRISTENSEN, J.; MARKS J.; SHIEBER, S. An empirical study of algorithms for point-feature label placement. **ACM Transactions on Graphics**. v. 14, n. 3, p. 203-232. Jul. 1995.
9. CORTES, M. B. S. Introdução à otimização. In: II Jornada de Estatística de Maringá. **Mini-curso: Introdução à otimização**. Maringá : UEM, Departamento de Estatística, 1999.
10. COUCLELIS, H. From cellular automata to urban models: new principles for model development and implementation. **Environment and planning B: planning and design**. v. 24, p. 165-174, 1997.
11. CRAMER, H. **Mathematical Methods of Statistics**". New York : John Wiley & Sons, 1971.
12. DAVIS, L. Adapting operator probabilities in Genetic Algorithms. **Proceedings of the Third International Conference on Genetic**

- Algorithms**. San Mateo, 61-69. 1989.
- 13.DAVIS, L. **Handbook of Genetic Algorithms**. Reissue edition. Stamford : International Thomson Publishing, 1996.
 - 14.DI CHIARA, B.; SORRENTINO, R.; STRAPPINI, M.; TARRICONE, L. Genetic optimization of radiobase-station sizing and location using a GIS-based framework: experimental validation. **IEEE Antennas and Propagation Society International Symposium**. v. 2, p.863-866. Jun. 2003.
 - 15.DRUCK, S.; CARVALHO, M. S.; CÂMARA, G.; MONTEIRO, A.V. M. (eds). **Análise Espacial de Dados Geográficos**. Brasília : EMBRAPA, 2004.
 - 16.EDELWEISS, N.; OLIVEIRA, J. P. M. **Modelagem de Aspectos Temporais de Sistemas de Informação**. Recife : UFPE-DI, 1994.
 - 17.ESHELMAN, L. J.; SHAFFER, D. J. Real-coded genetic algorithms and interval-schemata. In: WHITLEY, D. L. **Foundations of genetic algorithms 3**. San Mateo, CA: Morgan Kaufman, 1993, p.187-202.
 - 18.FISCHER, M. M.; LEUNG, Y. A genetic-algorithms based evolutionary computational neural network for modelling spatial interaction data. **Proceedings of 38th Congress of European Regional Science Association**. Set. 1998. <<http://www.ersa.org/ersaconfs/ersa98/papers/478.pdf>>. Visitado em 14/05/2004.
 - 19.GALVÃO, C. O.; VALENÇA, M. J. S. **Sistemas inteligentes: aplicações a recursos hídricos e sistemas ambientais**. Porto Alegre: Ed. Universidade/UFRGS/ABRH, 1999.
 - 20.GOLDBARG, M. C.; LUNA, H. P. L. **Otimização combinatória e programação linear: modelos e algoritmos**. Rio de Janeiro : Campus, 2000.
 - 21.GOLDBERG, D. E; HOLLAND, J. H. Genetic algorithms and machine learning: Introduction to the special issue on genetic algorithms. **Machine Learning**. 3, 1998.
 - 22.HERRERA, F.; LOZANO, M.; VERDEGAY, J. L. Tackling Real-Coded Genetic Algorithms: Operators and Tools for Behavioural Analysis. **Artificial Intelligence Review**. v. 12, n. 4, p. 265-319. 1998.
 - 23.HOLLAND, J. H. **Adaptation in natural and artificial systems**. Ann Arbor: University of Michigan Press, 1975.
 - 24.JHA, M. K.; McCALL, C.; SCHONFELD, P. Using GIS, genetic algorithms, and visualization in highway development. **Computer-Aided Civil and Infrastructure Engineering**. v. 16, n. 6, p. 399-414. 2001.
 - 25.KIRKPATRICK, S.; GELLAT, D. C.; VECCHI, M. P. Optimization by simulated annealing. **Science**. v. 220, n. 4598. p. 671-680, 1983.
 - 26.LAMBIN, E. F. Modeling Deforestation Processes - A Review. **Trees series**

B: Research Report. European Commission, Luxembourg, 1994.

27. LONGLEY, P. **Geocomputation: A Primer.** New York : John Wiley & Sons, 1998.
28. LORENA, L. A. N.; NARCISO, M. G. Uso de algoritmos genéticos em problemas de localização capacitada para alocação de recursos no campo e na cidade. **EMBRAPA - Comunicado Técnico.** v. 10, n.1. Dez. 2001.
29. LUCASIU, C. B.; KATEMAN, G. Applications of genetic algorithms in chemometrics. **Proceedings of the Third International Conference on Genetic Algorithms.** p. 170-176. 1989.
30. MAEDA, O.; NAKAMURA, M.; OMBUKI, B. M.; ONAGA, K. A genetic algorithm approach to vehicle routing problem with time deadlines in geographical information systems. **IEEE International Conference on Systems, Man and Cybernetics.** v. 2, p. 595-600. Out. 1999.
31. MALINVAUD, E. **Métodos estadísticos de la econometría.** Barcelona : Ariel, 1967.
32. MATTHEWS, K. B.; CRAW, S.; ELDER, S.; SIBBALD, A. R. Evaluating multi-objective land use planning tools using soft systems methods. **Proceedings of the 19th Annual Conference of the BCS Planning and Scheduling Special Interest Group.** p. 109-120. 2000.
33. MATTHEWS, K. B.; CRAW, S.; MACKENZIE, S. E.; SIBBALD, A. R. Applying genetic algorithms to land use planning. **Proceedings of the 18th Annual Conference of the BCS Planning and Scheduling Special Interest Group.** p. 109-115. 1999.
34. MENDES F^o, E. F. **Algoritmos Genéticos.** 2004. <<http://www.icmsc.sc.usp.br/~prico/gene1.html>>. Visitado em 20/07/2004.
35. METROPOLIS, W.; ROSENBLUTH, A.; ROSENBLUTH, M.; TELLER, A.; TELLER, E. Equation of state calculations by fast computing machines. **Journal of chemical physics.** v. 21, n. 6, p. 1087-1092, Jun. 1953.
36. MICHALEWICZ, Z. **Genetic algorithms + data structures = evolution programs.** 3.ed. Springer-Verlag, 1994.
37. NARCISO, M. G.; LORENA, L. A. N. Uso de algoritmos genéticos em sistema de apoio a decisão para alocação de recursos no campo e na cidade. **Revista Brasileira de Agroinformática.** v. 4, n. 2, p. 172-176. Mar. 2002.
38. NIX, H. A. A biogeographic analysis of Australian elapid snakes. In: **Atlas of Elapid Snakes of Australia.** Canberra : R. Longmore, 1986, p.5-15.
39. OPENSHAW, S. Building Automated Geographical Analysis and Explanation Machines. In: P. Longley (ed)., **Geocomputation: A Primer.** Chichester : John Wiley, 1998, p.94-115.

40. OPENSHAW, S.; ABRAHART, R. J. **Geocomputation. Proceedings of the 1st International Conference on Geocomputation.** Leeds : University of Leeds, p. 665-666, 1996.
41. OPENSHAW, S.; ABRAHART, R. J. **Geocomputation.** CRC Press. 2000.
42. OPENSHAW, S.; OPENSHAW, C. **Artificial intelligence in geography.** West Sussex : John Wiley & Sons Ltd. 1997.
43. PEDROSA, B. M. **Ambiente computacional para modelagem espacial dinâmica.** São José dos Campos, 2003. Tese (Doutorado em Computação Aplicada), Instituto Nacional de Pesquisas Espaciais.
44. PAYNE, K; STOCKWELL, D. R. B. **GARP Modelling System User's Guide and Technical Reference.** 2001. <<http://biodi.sdsc.edu/Doc/GARP/Manual/manual.html>>. Visitado em 15/04/2005.
45. REBELLO, F. R.; HAMACHER, S. Zoneamento e roteamento de veículos da coleta de correspondência dos correios usando algoritmos genéticos. **Anais do XIII ANPET.** 1999. <http://www.ind.puc-rio.br/artigos/art_hamacher_15.htm>. Visitado em 04/05/2004.
46. RONALD, S.; KIRKBY, S. Compound optimisation: solving transport and routing problems with a multi-chromosome genetic algorithm. **IEEE International Conference on Evolutionary Computation.** p. 365-370. Maio 1998.
47. RONALD, S.; KIRKBY, S. Genetic algorithms for geographical boundary assignment. **IEEE Evolutionary Computation Proceedings.** p. 235-240. Maio 1998.
48. SANTA CATARINA, A. **Model Breeder – Aplicação de um Algoritmo Genético numa ferramenta para análise de dados.** São José dos Campos, 4 p. Trabalho não publicado.
49. SANTOS, M. **A natureza do espaço: técnica e tempo, razão e emoção.** São Paulo : HUCITEC, 1996.
50. SIQUEIRA, M. Os algoritmos disponíveis no OpenModeller para modelagem de nicho: distância similaridade ambiental, Bioclim, CSM, GARP. In: OFICINA DE MODELAGEM DE DISTRIBUIÇÃO DE ESPÉCIE. <http://www.dpi.inpe.br/geoma_biod_mdea/WS_SJC/PPTS/algoritmos_marinez.pdf>. São José dos Campos : INPE, 2005.
51. SKOK, M.; SKRLEC, D.; KRAJCAR, S. Genetic algorithm and GIS enhanced long term planning of large link structured distribution systems. **IEEE Large Engineering Systems Conference on Power Engineering.** p. 55-60. Jun. 2002.
52. SKRLEC, D.; KRAJCAR, S.; PRIBICEVIC, B.; BLAGAJAC, S. Exploiting the power of genetic algorithm in optimization of distribution networks. **IEEE Electrotechnical Conference.** v. 3, p.1607-1610. Maio 1996.

53. STOCKWELL, D.; PETERS, D. **Spatial predictions using Genetic Algorithm for Rule-set Production**. 1993. <biodi.sdsc.edu/Symbiotik/Model/GARP/Doc/tutorial.html>. Visitado em 26/04/2005.
54. STOCKWELL, D.; PETERS, D. The GARP modeling system: problems and solutions to automated spatial prediction. **International Journal of Geographical Information Science**. v. 13, n. 2, p. 143-158. Mar. 1999.
55. TAN, K. C.; LEE, L. H.; ZHU, Q. L.; OU, K. Heuristic methods for vehicle routing problem with time windows. **Artificial Intelligence in Engineering**. v. 15, n. 3, p. 281-295. Jul. 2001.
56. TANURE, C. Z.; HAMACHER, S. Aplicação de algoritmo genético para um problema de distribuição dos correios. **Anais do XII ANPET**. 1998. <<http://www.ind.puc-rio.br/artigos/>>. Visitado em 04/05/2004.
57. TAT, C. W.; TAO, F. Using GIS and genetic algorithm in highway alignment optimization. **Proceedings of Intelligent Transportation Systems**. v. 2, p.1563-1567. Out. 2003.
58. UNIVERSITY OF KANSAS - Biodiversity Research Center. **DesktopGARP – Users Manual**. <<http://www.lifemapper.org/desktopgarp/Default.asp?Item=3&Lang=1>>. Visitado em 08/04/2005.
59. XIAO, N.; BENNETT, D. A.; ARMSTRONG, M. P. Solving spatio-temporal optimization problems with genetic algorithms: a case study of a bald cypress seed dispersal and seedling growth model. **4th International Conference on Integrating GIS and Environmental Modeling (GIS/EM4): Problems, Prospects and Research Needs**. 2000. <<http://www.colorado.edu/research/cires/banff/pubpapers/97>>. Visitado em 27/04/2005.
60. YAMAMOTO, M.; LORENA, L. A. N. A constructive genetic approach to point-feature cartographic label placement. **The Fifth Metaheuristics International Conference**. p. 841-847. Ago. 2003.
61. YEPES, I. **Uma incursão aos algoritmos genéticos**. 2004. <<http://www.geocities.com/igoryepes/>>. Visitado em 20/07/2004.