

ALGORITMO GENÉTICO GARP PARA MODELAGEM AMBIENTAL

Lucas Persona¹
Pedro Luiz Pizzigatti Corrêa²
Antonio Mauro Saraiva³

Resumo

Este trabalho apresenta o estudo realizado sobre Algoritmos Genéticos, que são baseados em um processo evolutivo na procura de soluções melhores adaptadas a uma situação. Dentre os Algoritmos Genéticos, uma atenção especial é dada ao Algoritmo Genético GARP (*Genetic Algorithm for Rule-set Production* – Algoritmo Genético para Produção de Conjunto de Regras). Formado por conjuntos de regras, o GARP segue os princípios dos Algoritmos Genéticos na busca de conjunto de regras que representem uma solução. Estas regras, utilizando informações sobre condições ambientais e disposição geográfica conhecida das espécies, podem ser utilizadas para obter modelos de distribuição de espécies em estudos sobre biodiversidade. Todo este estudo visa fornecer um conhecimento básico para a modelagem de uma arquitetura e uma implementação do algoritmo GARP em JAVA, que poderá ser utilizado como parte de um sistema de criação de modelos de predição ambiental, tornando possível prever, com certa segurança, qual será a organização e composição de diversos ecossistemas. Esta implementação tem por aplicação futura permitir um estudo da utilização do GARP em conjunto com uma base de dados de biodiversidade de abelhas brasileiras, para se identificar nichos ambientais das espécies estudadas. Este projeto de biodiversidade de abelhas brasileiras foi utilizado como referência aos estudos e obtenção de informação.

Palavras-chave: abelhas brasileiras, algoritmos genéticos, biodiversidade, GARP, modelagem ambiental, modelos de predição, nicho ambiental.

Abstract

This paper presents a research done on Genetic Algorithms, which are based on an evolutive process of searching the solutions that better fit on a situation. Among the Genetic Algorithms, a special attention is given to GARP (*Genetic Algorithm for Rule-set Production*). Made of a set of rules, GARP follows the basis from Genetic Algorithms in the search for a solution. These rules, using environment conditions and known geographical species distribution information, can be used to create predictive species distributions models for biodiversity research. The purpose of this research is to provide a knowledge base to create an architecture model and a JAVA implementation of the GARP algorithm, which could be used as part of a predictive species distribution creation system, making it possible to predict, with some level of assurance, what will be the organization and composition of several ecosystems. The purpose of the implementation is a research on GARP's usage altogether with a biodiversity database

¹ Graduado em Ciência da Computação pela Escola de Engenharia de Piracicaba. Pesquisa junto ao Departamento de Engenharia de Computação e Sistemas Digitais da POLI/USP

² Professor Doutor do Departamento de Engenharia de Computação e Sistemas Digitais da POLI/USP

³ Professor Livre-docente do Departamento de Engenharia de Computação e Sistemas Digitais da POLI/USP

of Brazilian bees, in order to identify ecological niches. This Brazilian bees' project was used as a reference and information source to this work.

Keywords: Brazilian bee, genetic algorithm, biodiversity, GARP, predictive modeling, prediction models, ecological niches.

1. Introdução

Os avanços da civilização moderna, nas mais diversas áreas, têm contribuído para mudanças no meio-ambiente e na relação entre seres vivos. Em função da aceleração do processo de extinção de várias espécies, decorrente de ações diretas ou indiretas dos seres humanos, torna-se necessário ser capaz de prever, com certa segurança, qual será a organização e composição de diversos ecossistemas.

Neste contexto, a utilização de Algoritmos Genéticos no processo de Modelagem Ambiental possibilita antecipar algumas mudanças como o deslocamento geográfico, extinção e adaptação de espécies. Segundo PETERSON, Modelagem Ambiental é “qualquer abstração do mundo natural com algum poder de prever eventos com base em princípios gerais” (PETERSON, 2002).

Diversos estudos estão sendo feitos em países como México, Canadá, Índia, Estados Unidos e também no Brasil, no sentido de identificar os tipos de efeitos das mudanças climáticas e seus impactos em diversos nichos ambientais (PETERSON, 2002). Estes nichos ambientais, ou ecológicos, é definido por PETERSON e VIEGLAIS como um conjunto de fatores ambientais que determinam onde uma espécie pode e/ou não pode manter populações (PETERSON;VIEGLAIS, 2001).

Um grande aliado destes trabalhos, o Algoritmo Genético GARP (*Genetic Algorithm for Rule-set Production* – Algoritmo Genético para Produção de Conjunto de Regras), utiliza-se de informações sobre condições ambientais, topográficas e disposição geográfica conhecida das espécies para prever modelos de distribuição das espécies.

Uma implementação do GARP em Java possibilitará uma maior abrangência no seu uso, devido à própria portabilidade da linguagem, além de permitir sua utilização através de um modelo de serviços distribuídos pela Internet.

Esta visão de modelagem, ao ser aplicada juntamente com informações de uma base de dados de biodiversidade de abelhas brasileiras do Projeto WebBee (WEBBEE, 2003), que tem por objetivo "organizar as informações e o conhecimento gerados nas pesquisas sobre os meliponídeos, e facilitar o acesso remoto a esses conteúdos, contribuindo para a sua difusão e estimulando o trabalho e a pesquisa cooperativos" (SARAIVA;IMPERATRIZ-FONSECA, 2002), torna possível identificar nichos ambientais das espécies estudadas.

2. Estudos e Métodos

2.1. Inteligência Artificial

Desde os primeiros cientistas da computação, como Alan Turing, John Von Neumann dentre outros, a idéia de se obter uma máquina dotada de certa inteligência era um objetivo comum. Neste sentido, as principais metodologias se baseavam em

princípios naturais, buscando aplicar os processos naturais às máquinas, levando os cientistas a campos de pesquisa além da computação e eletrônica tais como a filosofia, psicologia e biologia. BUCHANAN (2003) define uma linha do tempo da área de Inteligência Artificial que remonta da era de Aristóteles.

Entretanto, o termo “Inteligência Artificial” foi utilizado em 1956 por John McCarthy, também criador da Linguagem LISP – 1958 – no primeiro congresso dedicado ao tema: *Dartmouth Conference*. Desde então, a área de Inteligência Artificial permanece em franco crescimento e inicia o desdobramento da área em uma seqüência de novos campos como redes neurais, aprendizado de máquinas e também o campo da computação evolucionária.

2.2. Computação Evolucionária

Segundo MICHELL (1998), a idéia de aplicar um método evolucionário para solucionar problemas de engenharia já era estudada de forma independente por vários cientistas da computação nas décadas de 1960 e 1970. A idéia principal desses estudos era evoluir uma população de possíveis soluções a um problema, utilizando operadores inspirados na variação genética e na seleção natural.

Ainda conforme MICHELL (1998), no campo da computação evolucionária, surgiram três vertentes mais específicas no modo de se resolver problemas. Estratégias Evolutivas, por Rechenberg em 1965 e 1973, Programação Evolutiva, desenvolvida por Fogel, Owens e Walsh em 1966 e Algoritmos Genéticos, apresentados por John H. Holland em 1975.

2.3. Algoritmos Genéticos

Algoritmos Genéticos fazem parte do ramo de Computação Evolucionária, pertencente à área de Inteligência Artificial.

Inicialmente desenvolvidos na década de 60 por John H. Holland, os algoritmos genéticos baseiam-se em um processo evolutivo onde novos conjuntos de soluções são criados até que certas condições sejam atingidas, em outros termos, procuram-se soluções melhor adaptadas a uma situação.

Conforme HOLLAND, adaptação, em qualquer contexto, envolve uma modificação progressiva de alguma estrutura ou estruturas. Sugere que, em um processo de adaptação existem os modificadores das estruturas, responsáveis por criar novas estruturas possivelmente melhor adaptadas a uma situação. Esta situação é denominada por ele como “*the environment of the system undergoing adaptation* - o ambiente do sistema submetido à adaptação” (HOLLAND, 1992).

A ação dos modificadores nas estruturas corresponde ao plano adaptativo que, segundo HOLLAND (1992), determina quais estruturas aparecem em resposta ao ambiente, e o conjunto de estruturas alcançável, e, aplicando todas as seqüências possíveis de operadores, define os limites do domínio de ação do plano adaptativo.

Uma medida de aptidão, definida por HOLLAND (1992), identificará, a cada estágio do plano adaptativo, as estruturas mais adaptadas ao ambiente e que terão maior influência nas estruturas do próximo estágio do plano adaptativo.

Estes conceitos abordados pelos algoritmos genéticos são uma representação de processos naturais de adaptação das espécies. Considerando esta abordagem, os principais modificadores que estariam atuando sobre as estruturas são, na maioria das vezes, simulações dos processos de sobrevivência, reprodução, mutação e extinção dos indivíduos de uma população.

2.4. Algoritmo Genético GARP

O Algoritmo Genético GARP (*Genetic Algorithm for Rule-set Production* – Algoritmo Genético para Produção de Conjunto de Regras) baseia-se nas regras gerais dos algoritmos genéticos, mas as estruturas utilizadas são constituídas por conjuntos de regras. Estas regras são obtidas através de informações sobre condições ambientais, topográficas e disposição geográfica conhecida das espécies. Com elas, o GARP pode ser aplicado para se obter modelos de distribuição das espécies.

Criado por David Stockwell (STOCKWELL, 2003), o GARP tem como objetivo ser um método genérico e com desempenho confiável para analisar todos os dados e relacionamentos potenciais das espécies com o meio-ambiente. Os modelos do GARP são formados por um conjunto de regras, onde uma regra equivale a uma estrutura, ou indivíduo, de uma população conceituada nos termos dos Algoritmos Genéticos. Esta população de regras do GARP é então avaliada e novas populações são geradas de acordo com os conceitos de Algoritmos Genéticos, através dos operadores ou modificadores de estruturas.

De acordo com STOCKWELL e PAYNE (2003), o algoritmo utilizado pelo GARP é formado pelos seguintes passos:

i. *Criar uma População Inicial de Indivíduos:*

O conjunto inicial de informações biológicas contendo pontos de existência e/ou inexistência da espécie em estudo, de forma georeferenciada é utilizado como ponto de partida da identificação de uma população inicial de indivíduos. Estes indivíduos são estruturas formadas por regras que regem as condições nas quais a espécie em estudo teve a existência ou inexistência comprovada.

Para elaboração das regras, as informações relativas a condições ambientais e topográficas são recuperadas de acordo com a referência geográfica e temporal de cada registro apresentado nas informações biológicas.

Este conjunto de indivíduos pode ser estabelecido de forma específica ou mesmo recuperado de forma randômica, dentro do universo de dados informados.

A quantidade de indivíduos que esta população estará abrigando é estabelecida no início do processamento do algoritmo.

ii. *Avaliar a Adaptabilidade dos Indivíduos:*

Em uma população, sempre existirão os indivíduos mais representativos e mais aptos a determinadas situações.

Neste conceito, cada indivíduo formado por um conjunto de regras, tem sua aplicabilidade testada com base no conjunto de pontos de existência e/ou inexistência informados inicialmente.

Para isto, as regras de cada indivíduo são aplicadas ao conjunto de dados e a medida de adaptabilidade do indivíduo corresponde à respectiva porcentagem de dados que atendem àquele conjunto de regras.

Deste processo é possível obter uma classificação que represente quais indivíduos ou, conjunto de regras, são mais representativos na definição de existência ou ausência da espécie em estudo.

iii. Armazenar os Melhores Indivíduos:

Nesta atividade, o conjunto representando os melhores indivíduos é armazenado para referência posterior na possível definição de solução. Corresponde ao histórico das populações geradas que poderá ser utilizado para verificar o índice de alteração entre duas populações.

iv. Verificar Critério de Aceitação do Resultado:

Após a avaliação e armazenagem dos melhores indivíduos, uma verificação ocorre visando identificar o nível de alteração entre os melhores indivíduos da população obtida em uma iteração anterior e os melhores indivíduos da população atual. Este nível de alteração, em conjunto com um parâmetro inicial do algoritmo, define o grau de convergência aceitável como resultado no processamento do algoritmo. Caso este grau seja atingido, o processamento é encerrado e o conjunto armazenado é apresentado como solução.

v. Aplicar Modificadores de Estruturas à População:

Após a constatação que o nível de alteração e/ou adaptabilidade dos indivíduos não corresponde com a precisão desejada, a população será utilizada na criação de uma nova população, após a aplicação de diversos modificadores.

Estes modificadores, também chamados de operadores heurísticos (STOCKWELL; PAYNE, 2003), são responsáveis por alterar o espaço de busca do algoritmo e convergir a uma solução mais adequada. Os operadores mais aplicados são a união, o cruzamento e a mutação.

A partir do conjunto de indivíduos mais adaptados, conforme classificado pelo passo *ii*, os primeiros indivíduos serão persistidos na nova população e realizarão cruzamentos e uniões entre eles.

O operador de união, quando aplicado, cria a partir de dois indivíduos, um novo indivíduo contendo a união dos conjuntos de regras dos indivíduos originais.

No processo de cruzamento, dois indivíduos são utilizados e um ponto de corte comum a eles é estabelecido aleatoriamente. A partir deste ponto de corte, as regras são trocadas entre os indivíduos, obtendo indivíduos com nova organização de regras.

O processo de mutação age alterando uma regra de um indivíduo para novos valores definidos aleatoriamente. Ao agir sobre um indivíduo, uma de suas regras será escolhida para sofrer mutação, que pode afetar tanto a variável que define a regra, como o operador da regra (igualdade, desigualdade, etc.) ou constante valorada. Este processo é importante no quesito de encontrar novas áreas de busca que, possivelmente, poderiam trazer resultados mais aptos à solução desejada.

vi. *Reiniciar Iteração:*

Finalizando, o processo retorna ao passo *ii* para uma avaliação da adaptabilidade de cada indivíduo desta nova população.

Como critério de parada, também pode ser utilizado um contador de gerações que será incrementado após cada iteração.

2.5. Variáveis consideradas

As regras do GARP utilizam variáveis ambientais e informações de existência e inexistência das espécies para tornar possível identificar pontos georeferenciados e obter um modelo de distribuição de espécies.

Levando em consideração um levantamento dos dados mais comumente utilizados e informações disponíveis, as seguintes variáveis foram consideradas como aptas a definir parâmetros que identifiquem possibilidade de distribuição da espécie.

Tabela 1. Variáveis encontradas relevantes ao estudo.

Altitude	Umidade Relativa do Ar	Índice de Cobertura Vegetal
Precipitação	Pressão Atmosférica	Velocidade do Vento
Temperatura Média	Temperatura Mínima	Temperatura Máxima
Radiação Solar	Radiação Ultra-Violeta	Tipo de Vegetação

2.6. Recursos utilizados

O processo de obtenção das informações foi realizado através da utilização de algumas bibliotecas e, principalmente, do acesso a conteúdo disponibilizado pela Internet. O desenvolvimento do protótipo em si, não requeria maiores investimentos em hardware e software, utilizando ferramentas disponíveis gratuitamente.

Os dados necessários para validação do protótipo são provenientes do Projeto WebBee (WEBBEE, 2003) que tem por objetivo "organizar as informações e o conhecimento gerados nas pesquisas sobre os meliponídeos, e facilitar o acesso remoto a esses conteúdos, contribuindo para a sua difusão e estimulando trabalho e pesquisa cooperativos" (SARAIVA; IMPERATRIZ-FONSECA, 2002).

Estas informações tornam possível obter resultados mais precisos e reais no processo de geração dos modelos. Entretanto, sua falta não inviabiliza o trabalho. Os dados utilizados na validação do protótipo podem ser criados na forma de conjunto de valores de teste considerando as variáveis levantadas neste estudo.

Quanto às necessidades tecnológicas, para a continuidade dos trabalhos onde se adotará e implementará a modelagem proposta, torna-se necessário um levantamento que indique, de acordo com as aplicações da solução final, toda a estrutura necessária para efetivação do projeto.

3. Resultados e Discussão

3.1. Modelagem Proposta

A viabilização do modo de operação do GARP necessita uma modelagem específica e serviços agregados que, juntos, fornecerão suporte ao completo funcionamento do processo.

Este conjunto de serviços, proposto em Figura 1, sendo regido por um componente de coordenação (*Coordinator*) é inicializado através de uma requisição feita por um usuário ou cliente, de processamento e criação de um modelo de distribuição de determinada espécie, onde a apresentação do resultado pode ser feita na forma das regras existentes no modelo ou como um mapa gráfico identificando a distribuição da espécie.

Este usuário requisitante pode ser representado tanto por um ser humano através de interfaces de definição dos parâmetros de busca e configuração do algoritmo, como por um outro sistema ou serviço que necessite realizar o processamento e obter modelos de distribuição de muitas espécies.

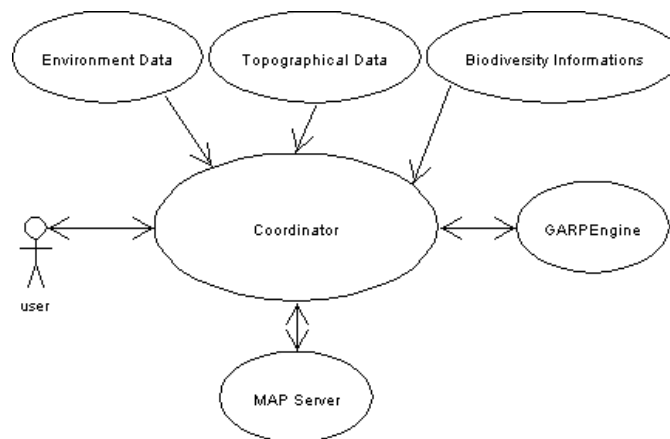


Figura 1. Modelo de Componentes sugerido como solução de aplicação baseada em serviços.

Após receber a requisição, o coordenador inicia então a seqüência de processamento solicitando informações de serviços que contenham, e disponibilizem, dados ambientais e topográficos. De acordo com a espécie definida na requisição inicial de processamento, serviços contendo informações de biodiversidade são pesquisados para se obter o conjunto conhecido de análises de existência ou inexistência da espécie requerida.

A partir deste momento, essas informações alimentam o processamento do algoritmo GARP, através do *GARPEngine*, que estará realizando todas as atividades anteriormente descritas, de acordo com o algoritmo especificado pelo GARP.

Para se realizar este processamento, um Diagrama de Classes (Figura 2) representa as classes básicas internas do serviço de *GARPEngine*. Esse serviço, regido pela classe *GARPEngine*, possui uma estrutura de agregação de classes que representam as entidades utilizadas pelo GARP.

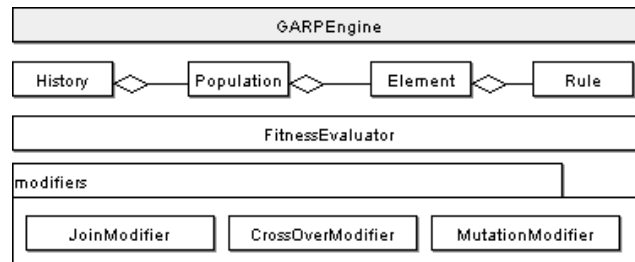


Figura 2. Modelo de Classes do serviço GARPEngine.

Inicialmente, a classe *History*, corresponde a um conjunto de populações, geradas e arquivadas representando uma linha do tempo das populações. Estas populações históricas podem ser utilizadas como parâmetro de interrupção do processamento do GARP, quando, após notar-se que certo conjunto de população não apresenta mudança significativa em seus elementos mais adaptados, é aceitável estabelecer que aqueles elementos são os que melhor definem a solução procurada.

Definindo uma população de indivíduos, a classe *Population* representa todos os indivíduos daquela população, ordenados de forma crescente de acordo com o valor de adaptabilidade de cada um.

Estes indivíduos, representados pela classe *Element*, agregam um conjunto de regras, representadas pela classe *Rule*, que estabelecem a existência ou inexistência de determinada espécie quando o conjunto de regras agregadas seja validado de forma positiva.

À parte da estrutura de representação dos elementos manipulados pelo algoritmo genético, a classe *FitnessEvaluator*, é responsável por calcular a adaptabilidade de um indivíduo de acordo com as informações assimiladas das coleções biológicas, verificando a validade daquele conjunto de regras e sua representatividade do conjunto de pontos de ocorrência da espécie.

Durante o processamento do algoritmo, modificadores dos indivíduos são utilizados na criação de novas populações. Estes modificadores são representados pelas classes *JoinModifier*, *CrossOverModifier* e *MutationModifier*.

Todo este conjunto, interno ao serviço do *GARPEngine*, é utilizado na execução do serviço que, depois de processado retorna as soluções relativas às regras encontradas e distribuição prevista da espécie ao *Coordinator*.

O próximo passo, antes de finalizar o processamento da requisição e retornar ao usuário, é identificar se o modo de retorno do modelo de regras encontrado será feito na forma de lista de regras, ou como representação gráfica das regras aplicadas a um mapa. No caso da necessidade de se retornar um mapa gráfico, é necessário que o *Coordinator* inicie um processo de consulta a um servidor de mapas (*Map Server*) para obter, e remeter ao usuário, uma representação gráfica do modelo de distribuição de espécies encontrado.

3.2. Implementação

Um protótipo da aplicação do algoritmo GARP foi implementado em Java, utilizando o SDK Java 2 Standard Edition versão 1.4.1 (SUN, 2003). Os módulos foram implementados conforme a prática de *Test-Driven Development* – Desenvolvimento Orientado por Testes (BECK, 2003), ou TDD.

De uma forma geral, segundo a prática TDD, todos os códigos são implementados para atender a testes desenvolvidos anteriormente. Estes testes, implementados neste projeto com o auxílio do *framework* JUnit (JUNIT, 2003), estarão representando os requisitos necessários ao código que será implementado, e estarão guiando processos posteriores de manutenção e suporte do código. Um outro aspecto importante é a possibilidade de se obter uma implementação enxuta, garantindo o correto funcionamento e objetividade de cada código.

O diagrama de classes do protótipo, seguindo a modelagem proposta na Figura 2, representa as classes e respectivas dependências como apresentado na Figura 3:

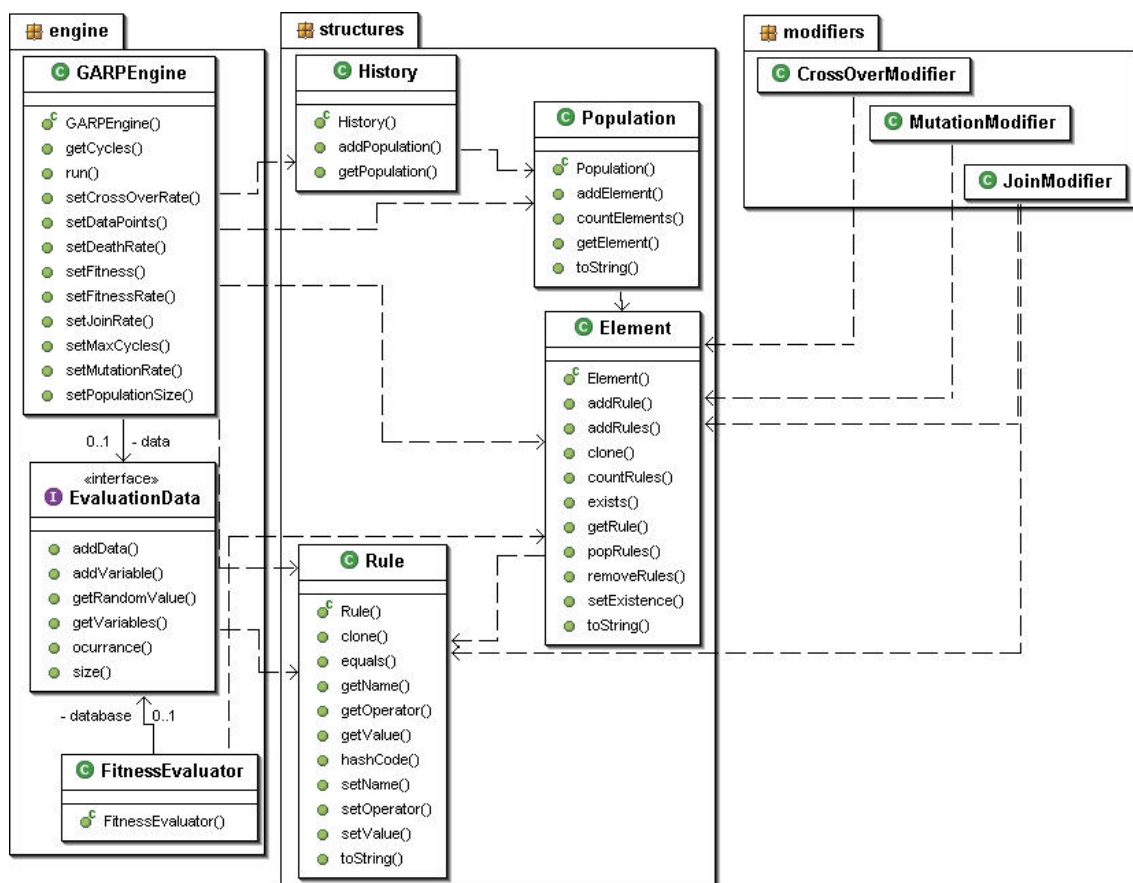


Figura 3. Diagrama de Classes do protótipo do GARP.

O *GARPEngine* implementado no protótipo apresenta um conjunto de parâmetros de configuração do algoritmo relacionados à inicialização e critérios de parada de processamento. As configurações de inicialização são as seguintes:

Taxa de aplicação do operador de cruzamento (*CrossOverRate*): representa a probabilidade de ocorrer um cruzamento no momento de aplicação de operador heurístico.

Taxa de aplicação do operador de união (JoinRate): representa a probabilidade de ocorrer uma união no momento de aplicação de operador heurístico.

Taxa de aplicação do operador de mutação (MutationRate): representa a probabilidade de ocorrer uma mutação no momento de aplicação de operador heurístico.

Tamanho da população (PopulationSize): define a quantidade de indivíduos que cada população poderá ter.

Taxa de mortalidade (DeathRate): representa o índice de mortalidade em cada geração. Esta taxa define a porção renovável de indivíduos a cada nova geração.

Uma vez iniciado o processamento, o protótipo entra em laço de execução até que pelo menos uma das condições de parada seja encontrada. Estas condições de parada são as seguintes:

Quantidade máxima de gerações (MaxCycles): esta condição de parada restringe a quantidade máxima de novas populações geradas.

Índice de adaptabilidade (Fitness): representa o índice mínimo aceitável de adaptabilidade de um indivíduo como solução do modelo. Esta configuração será sempre utilizada em conjunto com a Taxa de análise de adaptabilidade.

Taxa de análise de adaptabilidade (FitnessRate): representa o percentual de indivíduos que deverão atender à configuração de Índice de adaptabilidade para serem considerados aceitáveis.

3.3. Resultado do Protótipo

A execução do protótipo permitiu constatar a convergência que existe do GARP na obtenção de conjuntos de regras que possuem alto índice de adaptabilidade.

Baseado em um conjunto de dados gerados manualmente de forma aleatória, o protótipo foi executado com os seguintes critérios de inicialização e parada:

CrossOverRate: 60%	JoinRate: 30%	MutationRate: 10%
PopulationSize: 20	DeathRate: 50%	
MaxCycles: 50	Fitness: 90%	FitnessRate: 50%

Conforme o algoritmo GARP, uma população inicial de regras foi gerada a partir do conjunto de dados inicial. O segmento XML a seguir representa parte desta população inicial:

```
<population>
  <element fitness="0.16666667">
    <rule name="altitude" operator="3" values="1030" />
    <rule name="max_temp" operator="3" values="41.7" />
  </element>
  <element fitness="0.0">
    <rule name="altitude" operator="2" values="9100" />
    <rule name="altitude" operator="1" values="2200" />
    <rule name="umidade" operator="2" values="0.36" />
  </element>
</population>
```

```

        <rule name="altitude" operator="3" values="1530" />
        <rule name="max_temp" operator="3" values="33.9" />
        <rule name="umidade" operator="1" values="0.23" />
        <rule name="altitude" operator="3" values="2310" />
        <rule name="min_temp" operator="3" values="3.78" />
        <rule name="min_temp" operator="2" values="13.3" />
        <rule name="max_temp" operator="3" values="42.3" />
        <rule name="min_temp" operator="2" values="9.7" />
        <rule name="min_temp" operator="3" values="16.2" />
    </element>
    <element fitness="0.0">
        <rule name="umidade" operator="3" values="0.22" />
        <rule name="umidade" operator="3" values="0.26" />
        <rule name="altitude" operator="3" values="1800" />
        <rule name="min_temp" operator="3" values="25.6" />
        <rule name="altitude" operator="2" values="1000" />
    </element>
    <element fitness="0.0">
        <rule name="min_temp" operator="2" values="12.7" />
        <rule name="altitude" operator="2" values="3450" />
        <rule name="max_temp" operator="1" values="43.2" />
        <rule name="umidade" operator="3" values="0.27" />
        <rule name="altitude" operator="2" values="2170" />
    </element>
    (...)
</population>

```

Após o processamento do algoritmo, a população resultante é a seguinte, representada por uma parte do XML:

```

<population>
    <element fitness="0.8333333">
        <rule name="min_temp" operator="2" values="20.5" />
        <rule name="min_temp" operator="2" values="25.6" />
        <rule name="altitude" operator="3" values="20.5" />
        <rule name="min_temp" operator="2" values="2310" />
        <rule name="min_temp" operator="2" values="20.5" />
    </element>
    <element fitness="0.8333333">
        <rule name="min_temp" operator="2" values="2310" />
        <rule name="min_temp" operator="2" values="25.6" />
        <rule name="altitude" operator="3" values="20.5" />
        <rule name="min_temp" operator="2" values="2310" />
        <rule name="min_temp" operator="2" values="20.5" />
    </element>
    (...)
</population>

```

4. Conclusões

A aplicação de Algoritmos Genéticos a este domínio de modelagem ambiental surge como alternativa a outros algoritmos de busca que poderiam inviabilizar a criação de um modelo de predição. Esta inviabilidade pode ocorrer devido à grande quantidade de variáveis necessárias para a criação dos modelos e também à falta de informações no campo de atuação definido. Com a utilização de algoritmos genéticos, esta falta de informações não impossibilita a busca por uma solução.

Outro aspecto resultante de algoritmos de busca alternativos é a possibilidade dos resultados se restringirem a mínimos e máximos locais, não tendo então uma melhor representação da realidade. Este problema é mais bem coberto pelo algoritmo genético através da criação de novas frentes de buscas após a aplicação dos modificadores de estruturas.

A implementação do modelo estruturado e sugerido permite disponibilizar serviços de criação de modelos de nichos ecológicos com um melhor aproveitamento dos dados, possibilitando avanços importantes nas pesquisas em Biodiversidade.

Uma futura linha de pesquisa pode surgir com a implementação deste modelo em conjunto com técnicas de processamento paralelo para agilizar e permitir buscas mais extensas com melhor grau de precisão.

Referência Bibliográficas

BECK, K. **Test Driven Development: By Example**. Addison-Wesley Pub. Co., 2003.

BUCHANAN, B.G. American Association for Artificial Intelligence. **History of AI**. Disponível em: <<http://www.aaai.org/AITopics/bbhist.html>>. Acesso em: 10 ago. 2003.

DESKTOPGARP. Disponível em: <<http://www.lifemapper.org/desktopgarp>>. Acesso em: 3 mai. 2003.

HOLLAND, J.H. **Adaptation in Natural and Artificial Systems**. Cambridge, Massachusetts: University of Michigan Press. MIT Press., 1992.

HOLLAND, J.H. Genetic Algorithms. **Scientific American**. New York, v. 267, n. 1, p. 44-50, jul. 1992.

JUNIT. JUnit, Testing Resources for Extreme Programming. Disponível em: <<http://www.junit.org>>. Acesso em: 30 out. 2003.

LIFEMAPPER. Informatics Biodiversity Research Center University of Kansas. Lawrence, KS. Disponível em: <<http://www.lifemapper.org>>. Acesso em: 14 jun. 2003.

MICHELL, M. **An Introduction to Genetic Algorithms (Complex Adaptive Systems)**. MIT Press., 1998

PETERSON, A.T. Predicting Species' Geographic Distributions based on Ecological Niche Modeling. **The Condor**. v. 103, n. 3, p. 599-605, ago. 2001.

PETERSON, A.T. Simuladores do Futuro. **Revista Pesquisa FAPESP**. São Paulo, n.80, p. 32-35, out. 2002.

PETERSON, A.T.; VIEGLAIS, D.A. Predicting Species Invasions Using Ecological Niche Modeling: New Approaches from Bioinformatics Attack a Pressing Problem. **Bioscience**. v. 51, p. 363-371, mai. 2001.

SARAIVA, A.M.; IMPERATRIZ-FONSECA, V.L. WebBee: Uma rede de informações sobre biodiversidade brasileira em abelhas nativas. **Proc. 5o. Encontro sobre Abelhas**. Ribeirão Preto, São Paulo. FFCLRP/FMRP/USP, 2002. p. 108-113.

STOCKWELL, D.R.B.; PAYNE, K. **GARP Modelling System User's Guide and Technical Reference**. San Diego Supercomputer Center. Disponível em: <<http://biodi.sdsc.edu/Doc/GARP/Manual/manual.html>>. Acesso em: 3 mai. 2003.

STOCKWELL, D.R.B. **Genetic algorithms for species distribution modeling**. Disponível em: <<http://biodi.sdsc.edu/~davids/Project/SGER/GAS/outline.html>>. Acesso em: 4 mai. 2003.

SUN. The Source for Java Technology. Disponível em: <<http://java.sun.com>>. Acesso em: 3 mai. 2003.

WEBBEE. **Projeto WebBee**. Universidade de São Paulo. São Paulo, SP. Disponível em: <<http://www.webbee.org.br>>. Acesso em: 12 abr. 2003.

WHYWHERE. Biodiversity Insight Systems. San Diego Supercomputer Center. Disponível em: <http://biodi.sdsc.edu/ww_home.html>. Acesso em: 3 mai. 2003.