

## Girafas: Uma Ferramenta Computacional para Apoio no Ensino de Algoritmos Genéticos

Flávio Roberto Santos<sup>1</sup>, Flavio V D de Figueiredo<sup>2</sup>, Vinicius F C Florentino<sup>3</sup>,  
Joseana Macêdo Fechine<sup>4</sup>

<sup>1,2,3,4</sup>Departamento de Sistemas e Computação

Centro de Engenharia Elétrica e Informática

Universidade Federal de Campina Grande

flaviors@dsc.ufcg.edu.br, flaviov@lsd.ufcg.edu.br,  
vinicius@lsd.ufcg.edu.br, joseana@dsc.ufcg.edu.br

**Resumo.** Algoritmos Genéticos utilizam o conceito de evolução observado na natureza para solucionar problemas computacionais de forma mais rápida do que com o uso de algoritmos tradicionais. Os problemas comuns que utilizam algoritmos genéticos são os de escalonamento e alocação. Atualmente, esta técnica é utilizada em diversas áreas, inclusive fora do âmbito da computação. Este trabalho descreve a construção de uma ferramenta, Girafas, com fins didáticos que visa aplicar e validar o uso de algoritmos genéticos em um ambiente lúdico. A ferramenta utiliza um cenário virtual composto por girafas e árvores, no qual é observada a evolução das girafas no tempo, a partir da manipulação das variáveis que modelam o ambiente.

### 1 Introdução

Os Algoritmos Genéticos (AG) são uma família de modelos computacionais inspirados na evolução que utilizam esse conceito para solucionar, por exemplo, problemas computacionais que demandariam muito tempo com o uso de algoritmos tradicionais. Os problemas mais comuns que utilizam algoritmos genéticos são os de escalonamento e alocação. Atualmente, esta técnica também está sendo usada em diversas áreas, inclusive fora do âmbito da computação (GOUNARES e SIKCHI, 2002; RUSSEL and NORVIG, 2003).

Dentro desse contexto, o objetivo do trabalho é o desenvolvimento de uma ferramenta computacional para fins didáticos, denominada Girafas, que visa à aplicação dos conceitos relacionados aos Algoritmos Genéticos (RUSSEL and NORVIG, 2003).

A ferramenta Girafas possibilita análises comparativas de diferentes aspectos da evolução de uma população, utilizando diferentes tipos de heurísticas para tal. O sistema permite ainda que usuários criem suas próprias heurísticas em Java e carreguem no sistema. Essa flexibilidade é um artifício bastante valioso no auxílio à compreensão de conceitos da Inteligência Artificial no âmbito dos Algoritmos Genéticos.

Além desse nível de interação entre o usuário e o sistema, será possível notar, a partir de comparações de resultados, a importância da escolha de uma boa heurística para se atingir o resultado esperado na população final. Essas funcionalidades possibilitarão ao Girafas atingir os objetivos aos quais se propõe, ser uma ferramenta

computacional, para fins didáticos, que possibilita ao usuário a alteração da configuração das entradas, de forma a avaliar o comportamento do Algoritmo Genético (SMITH E SZATHMÁRY, 1999).

As demais seções deste artigo estão organizadas conforme descrição a seguir. Na Seção 2 é feita uma breve descrição dos conceitos relacionados aos Algoritmos Genéticos. Em seguida, na Seção 3 explica-se como foi construída a ferramenta. Na Seção 4 é apresentada sua arquitetura. As heurísticas utilizadas e os resultados são apresentados na Seção 5. Por fim, as considerações finais sobre o trabalho são apresentadas na Seção 6.

## 2 Algoritmos Genéticos

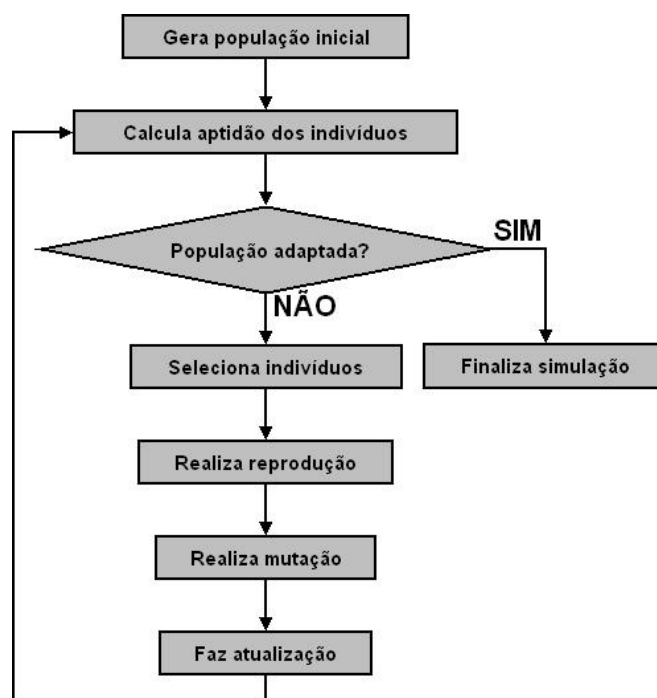
Algoritmos Genéticos (AG) constituem uma técnica da área da Inteligência Artificial e são algoritmos matemáticos inspirados nos mecanismos de evolução natural e recombinação genética. Os AG fornecem um mecanismo de busca adaptativa que se baseia no princípio Darwiniano de reprodução e sobrevivência dos mais aptos (RUSSEL and NORVIG, 2003; PACHECO, 2006; COELHO ET. AL., 2002; MITCHELL, 1996).

De acordo com a teoria de C. Darwin, o princípio de seleção privilegia os indivíduos mais aptos com maior longevidade e, portanto, com maior probabilidade de reprodução. Indivíduos com mais descendentes têm mais chance de perpetuar seus códigos genéticos nas próximas gerações. Tais códigos genéticos constituem a identidade de cada indivíduo e estão representados nos cromossomos. Estes princípios são utilizados na construção de algoritmos computacionais que buscam uma melhor solução para um determinado problema, através da evolução de populações de soluções codificadas a partir de cromossomos artificiais (PACHECO, 2006).

"Quanto melhor um indivíduo se adaptar ao seu meio ambiente, maior será sua chance de sobreviver e gerar descendentes": este é o conceito básico da evolução genética biológica (CARVALHO, 2006). Assim como acontece no meio ambiente, em um AG existe um grupo de indivíduos que competem entre si para garantir a própria sobrevivência e para assegurar que suas características sejam passadas adiante através da prole. Nessa analogia, cada indivíduo do AG é, de fato, uma solução candidata para o problema em questão, que, por sua vez, faz o papel do próprio meio-ambiente, já que estabelece os critérios que permitem avaliar se um indivíduo/solução é adaptado ou não.

Na Figura 1 é apresentada a descrição geral de um Algoritmo Genético (GOUNARES e SIKCHI, 2002).

O primeiro passo na definição de um AG é a escolha de uma maneira para representar as potenciais soluções. Esse processo é conhecido como *codificação*. O próximo passo seria definir uma maneira de avaliar os indivíduos, isto é, analisar quantitativamente o seu grau de adequação ao "meio". A medida numérica de qualidade de um indivíduo é chamada de *aptidão*.



**Figura 1 Descrição geral do Algoritmo Genético.**

Após esses passos, pode-se iniciar o processo iterativo descrito na Figura 1. Num primeiro momento, deve-se *inicializar* a população de indivíduos. Em geral, essa inicialização é feita de maneira aleatória.

A seguir, deve-se selecionar indivíduos para serem recombinados. É importante observar que a *seleção* não é feita de maneira determinística. Qualquer indivíduo pode ser escolhido, contando cada um com uma probabilidade proporcional à sua aptidão. Quanto mais “apto” for o indivíduo, maior a sua chance de ser selecionado. Aplica-se então o operador de *recombinação*, em geral chamado de *crossover*. O que a recombinação faz, em termos simples, é combinar o material genético de dois ou mais indivíduos para gerar descendentes.

Os indivíduos gerados pela recombinação devem então ser submetidos à *mutação*. A mutação, diferentemente do *crossover*, é um operador unário. Ela é aplicada sobre um único indivíduo e provoca uma modificação, em geral pequena. Esse processo garante (ou pelo menos torna teoricamente possível) o surgimento de características novas, inexistentes até então em uma população finita.

As novas soluções candidatas geradas a partir da recombinação e da mutação constituem a próxima geração. Não é difícil perceber que os indivíduos desse novo conjunto tendem a ser melhores do que aqueles que lhe deram origem, uma vez que a seleção valoriza os indivíduos mais aptos. Se esse processo for sucessivamente repetido, existe uma boa chance de que a solução ótima seja alcançada embora, em alguns casos, uma solução razoavelmente boa seja mais do que satisfatório. Uma vez encontrada a solução, o processo deve ser interrompido. Para tal, é necessário definir um *critério de parada*. Isso pode ser feito de várias maneiras. Uma delas consiste em estabelecer um número limite de gerações/iterações (BARRETO, 2003; RUSSEL and NORVIG, 2003).

Na Figura 2 é apresentado um exemplo de procedimento para implementação de um AG. Durante esse processo, os melhores indivíduos, e alguns dados estatísticos,

podem ser coletados e armazenados para avaliação (MAIA e MICHELAN, 2006).

```
Algoritmo_Genetico() {  
    t = 0;  
    inicia_população (P, t);  
    avaliação (P, t);  
    repita até (t = d) {  
        t = t + 1;  
        seleção_dos_pais (P, t);  
        recombinação (P, t);  
        mutação (P, t);  
        avaliação (P, t);  
        sobrevivem (P, t);  
    }  
}  
  
t - tempo atual;  
d - tempo determinado para finalizar o algoritmo;  
P - população.
```

**Figura 2 Procedimento para implementação de um Algoritmo Genético.**

### 3 Descrição do Girafas

O Girafas é um sistema para representação e estudo de Algoritmos Genéticos desenvolvido em Java (SUN MICROSYSTEMS, 2006). Para tanto, foi criado um ambiente virtual composto por girafas e árvores. Os animais dispostos no sistema precisam se adaptar ao meio para conseguir sobreviver. A adaptação consiste em variar os comprimentos dos pescoços das girafas. A dificuldade imposta decorre da variação das alturas dos galhos que contêm o alimento. Dessa forma, apenas as girafas que alcançam o topo das árvores conseguem se alimentar.

O Girafas foi concebido da seguinte forma (conforme Figura 2): inicialmente, uma população de girafas é criada de forma aleatória e inserida em um ambiente com árvores. Partindo-se dessa população inicial, é realizada uma seqüência de simulações no sistema. Cada simulação executada, também conhecida como Era, considera as características da simulação anterior para gerar um histórico da evolução dos indivíduos, satisfazendo o procedimento já descrito na Figura 1.

Uma Era possui um tempo pré-definido pelo usuário do sistema. Quanto menor for esse tempo, menor será a adaptação das girafas ao ambiente. Durante uma Era os animais caminham em direções aleatórias, podendo apenas se alimentar quando alcançam as folhas das árvores. É importante ressaltar, que nessa ferramenta foi imposto também um limite acima da árvore, fazendo com que uma girafa com um pescoço muito grande não consiga se alimentar. Isso foi feito apenas para melhor visualização dos resultados, uma vez que uma população ideal (adaptada ao ambiente) deve ser finalizada com todos os seus indivíduos com a mesma altura dos galhos.

### 4 Arquitetura

O sistema é constituído, basicamente, por três entidades principais: *GirafaObject*, *ArvoreObject* e *WorldLogic*.

A primeira e a segunda entidade são habitantes da terceira. As instâncias de *GirafaObject* possuem características como tamanho do pescoço, direção, satisfação e

dois outros atributos que representam quanto de satisfação a girafa ganha ao comer e quanto perde por não estar comendo.

As instâncias de *ArvoreObject* são objetos mais simples e a única informação que possuem é a sua altura. O mundo simulado pelo Girafas assume que esses recursos são inesgotáveis.

A função da terceira entidade é manter o histórico das populações e aplicar as heurísticas de evolução. Para isso, o *WorldLogic* mantém o controle do tempo decorrido na Era atual e, quando o período de tempo definido pelo usuário for atingido, uma nova população é criada.

Um ponto bastante relevante do trabalho refere-se à forma de avaliação da aptidão das girafas. Ao invés de aplicar uma função *fitness* nos indivíduos, eles interagem e reagem ao ambiente aumentando ou diminuindo a satisfação à medida que se alimentam ou não dos galhos das árvores. Dessa forma, é possível obter os mais adaptados no sistema simplesmente consultando os seus níveis de satisfação.

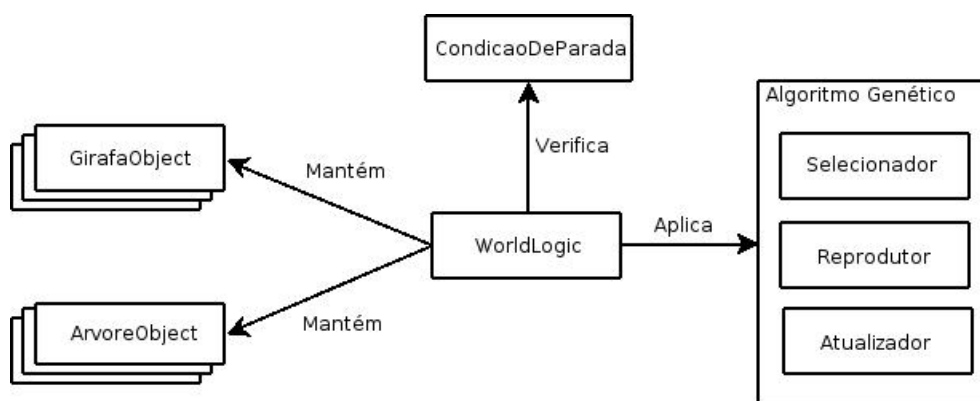
No Girafas o AG é dividido em três entidades principais:

- **Selecionador:** Encarrega-se de escolher as girafas mais adaptadas para reproduzir. Essa entidade considera o parâmetro da função de *fitness* para agir.
- **Reprodutor:** Aplica um algoritmo de cruzamento entre as girafas selecionadas pelo *Selecionador*. A idéia é gerar novos indivíduos com características herdadas dos pais, possibilitando a evolução da população.
- **Atualizador:** Define a população da nova geração de girafas de acordo com a população atual e os filhos gerados pelo *Reprodutor*.

Dividindo o AG nestas três entidades, obtém-se mais flexibilidade para a implementação de novos AG, pois é possível reutilizar entidades já criadas combinadas com novas instâncias e, em seguida, inseri-las no sistema.

Um outro aspecto importante do Girafas é a condição de parada. Uma simulação continua sendo executada até que uma certa condição seja atingida ou um dado número de populações seja gerado. Experimentos distintos provavelmente classificam de forma diferente quando é que uma população está adaptada. Assim, é a entidade *CondicaoDeParada* que se encarrega de verificar se a população está adaptada e encerrar uma simulação.

A interação entre as entidades descritas é mostrada na Figura 3 e é dada da seguinte forma: A entidade *WorldLogic* mantém as coleções de *GirafaObject* e *ArvoreObject*, e ao final de cada Era ela verifica se a população está adaptada de acordo com a *CondicaoDeParada*. Em caso negativo, o AG é aplicado na população e inicia-se uma nova Era.



**Figura 3 Arquitetura do Girafas.**

## 5 Análise do Girafas

Para validar o trabalho, dois AG foram implementados e analisados. Cada um deles composto de um *Selecionador*, um *Reprodutor* e um *Atualizador*.

Para possibilitar a análise mais abrangente de diferentes aspectos de uma população, foram implementadas duas heurísticas diferentes para cada entidade.

Através da implementação dessas heurísticas, é possível mostrar não só a importância da escolha de uma boa heurística, como também fazer comparações e análises dos algoritmos. A descrição do comportamento de cada algoritmo implementado é apresentada a seguir.

### 5.1 Heurísticas Utilizadas

As duas heurísticas utilizadas proporcionaram a implementação de dois algoritmos genéticos denominados Algoritmo Genético 1 (AG1) e Algoritmo Genético 2 (AG2).

#### 5.1.1 Algoritmo Genético 1 (AG1)

A heurística utilizada nesta implementação consiste de: para geração de uma nova população são selecionadas 50% do total de girafas da população atual. Os cromossomos das girafas selecionadas sofrem mutação invertendo seus bits (Ex: cromossomo 01 vira 10). As girafas selecionadas e seus filhos gerados irão formar a próxima população.

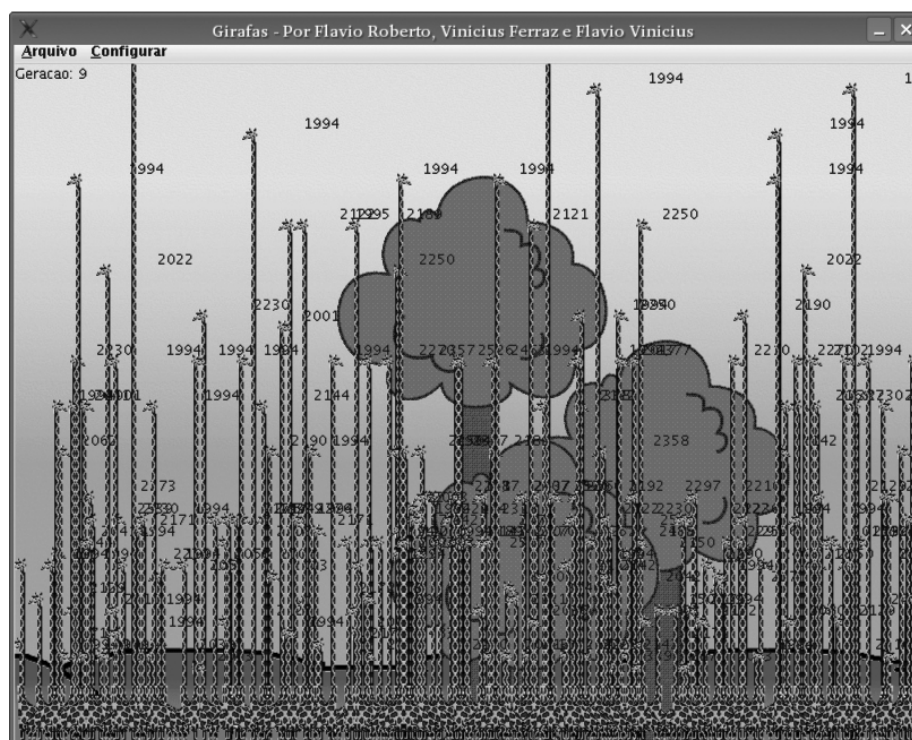
#### 5.1.2 Algoritmo Genético 2 (AG2)

Para geração de uma nova população, do total de girafas, as 20% melhores são selecionadas e duas a duas são separadas para o cruzamento. No cruzamento, os cromossomos são divididos em pontos aleatórios, e a troca de informação entre eles ocorre, gerando-se assim os novos filhos. Estes podem sofrer, ou não, mutação com uma probabilidade de 1%. A nova população será constituída das 80% melhores girafas da população anterior e dos filhos gerados no cruzamento.

### 5.2 Apresentação dos Resultados

A análise da evolução de um AG pode ser feita de duas formas:

- Forma gráfica: observando como a altura das girafas converge para a altura da copa das árvores (Figura 4).
- Forma textual: com gravação de log da geração atual, tempo e número de girafas que atendem à condição de parada.



**Figura 4 Girafas em Execução.**

A condição de parada nos testes é de que 100% das girafas estejam com a cabeça na região horizontal da copa de qualquer árvore do ambiente, ou quando 100 gerações são atingidas.

Ambos os algoritmos foram executados 80 vezes. A quantidade média de gerações necessárias para que a condição de parada fosse atingida foi o parâmetro utilizado para definir a eficiência do AG.

É possível observar, a partir da Tabela 1, que o AG1, que é totalmente aleatório, sempre teve a quantidade máxima de gerações atingidas, diferentemente do AG2.

**Tabela 1 Análise Comparativa dos Algoritmos Genéticos AG1 e AG2.**

Algoritmo Genético	Número Médio de Gerações
AG1	100,0
AG2	13,7

Verifica-se que a heurística utilizada no AG2, mais bem definida, foi superior à heurística utilizada no AG1.

Os resultados obtidos mostram, portanto, que o Girafas permite a análise da eficiência do algoritmo para diferentes tipos de heurísticas, possibilitando ao usuário não apenas avaliar os resultados obtidos, como também adicionar novos testes, bastando para isso construir novas heurísticas.

## 6 Considerações Finais

O Girafas cumpre o papel ao qual se propõe, se constituindo em uma ferramenta computacional que poderá ser utilizada para fins didáticos, possibilitando aos professores e estudantes da área de Inteligência Artificial, mais especificamente da área de Algoritmos Genéticos, implementar e testar diversas formas de implementação com facilidade, a partir do uso de heurísticas.

Em virtude do uso de uma ferramenta inspirada em um modelo tradicional contendo girafas, a evolução da população, em resposta à sua heurística, é apresentada ao usuário de maneira fácil e intuitiva. Desse modo, ele pode acompanhar facilmente a convergência das alturas das girafas em relação às árvores do ambiente e ainda comparar a heurística utilizada na sua implementação com as demais heurísticas.

Como trabalho futuro, pretende-se adicionar a geração de relatórios e gráficos das simulações sendo executadas no Girafas, facilitando ainda mais o estudo de Algoritmos Genéticos.

Pretende-se também tornar o ambiente de simulação do Girafas mais complexo, como por exemplo ganho de satisfação variado de acordo com diversos tipos de árvores e relevos do terreno. Dessa forma, a aptidão da população seria função de outros fatores (além da altura dos indivíduos) como localização da girafa e tipos de árvores que podem ser comidas.

## Referências

BARRETO, A. da M. S. (2003), “Uma Introdução aos Algoritmos Genéticos” <http://www.coc.ufrj.br/~andremsb/textos.htm>

COELHO, C. A. C.; VELDHUIZEN, D. A. V. and LAMONT, G. B. (2002) “Evolutionary Algorithms for Solving Multi-Objective Problems”. Kluwer Academic Publishers, 5nd edition.

DE CARVALHO, A. D. de L. F. (2006) “Algoritmos Genéticos. <http://www.icmc.usp.br/~andre/research/genetic/index.htm>”, Setembro.

GOUNARES, A. and SIKCHI, P. (2001) “Adaptive problem solving method and apparatus utilizing evolutionary computation techniques”. U.S. Patent.

MAIA, A. C. P. and MICHELAN, R. (2006) “Algoritmos Genéticos”, <http://www.din.uem.br/ia/geneticos>, Setembro.

MITCHELL, M (1996)., “An Introduction to Genetic Algorithms”. MIT Press.

PACHECO, M. A. C. (2006) “Algoritmos Genéticos Princípios e Aplicações”, ICA: Laboratório de Inteligência Computacional Aplicada, <http://www.ICA.ele.puc-rio.br>, Setembro.

RUSSEL, S. and NORVIG, P. (2003) “Artificial Intelligence – A Modern Approach. Prentice-Hall”, 2nd edition.

SMITH, J. M. and SZATHMÁRY, E. (1999) “The origins of life: From the birth of life to the origin of language”. Oxford University Press.

SUN MICROSYSTEMS (2006) “Sun Java Technology”. <http://java.sun.com>.