

# 1. INTRODUÇÃO

Computação Evolucionária (CE) é um ramo de pesquisa emergente da Inteligência Artificial que propõe um novo paradigma para solução de problemas inspirado na Seleção Natural (Darwin 1859).

A Computação Evolucionária compreende um conjunto de técnicas de busca e otimização inspiradas na evolução natural das espécies. Desta forma, cria-se uma população de indivíduos que vão reproduzir e competir pela sobrevivência. Os melhores sobrevivem e transferem suas características a novas gerações. As técnicas atualmente incluem (Banzhaf 1998): Programação Evolucionária, Estratégias Evolucionárias, Algoritmos Genéticos e Programação Genética. Estes métodos estão sendo utilizados, cada vez mais, pela comunidade de inteligência artificial para obter modelos de inteligência computacional (Barreto 1997).

Algoritmos Genéticos (AG) e Programação Genética (PG) são as duas principais frentes de pesquisa em CE. Os Algoritmos Genéticos (AG) foram concebidos em 1960 por John Holland (Holland 1975), com o objetivo inicial de estudar os fenômenos relacionados à adaptação das espécies e da seleção natural que ocorre na natureza (Darwin 1859), bem como desenvolver uma maneira de incorporar estes conceitos aos computadores (Mitchell 1997).

Os AGs possuem uma larga aplicação em muitas áreas científicas, entre as quais podem ser citados problemas de otimização de soluções, aprendizado de máquina, desenvolvimento de estratégias e fórmulas matemáticas, análise de modelos econômicos, problemas de engenharia, diversas aplicações na Biologia como simulação de bactérias, sistemas imunológicos, ecossistemas, descoberta de formato e propriedades de moléculas orgânicas (Mitchell 1997).

Programação Genética (PG) é uma técnica de geração automática de programas de computador criada por John Koza (Koza 1992), inspirada na teoria de AGs de Holland. Em PG é possível criar e manipular software geneticamente, aplicando conceitos herdados da Biologia para gerar programas de computador automaticamente.

A diferença essencial entre AG e PG é que em PG as estruturas manipuladas são bastante mais complexas, assim como várias das operações realizadas pelo algoritmo. Ambas as técnicas compartilham a mesma base teórica, inspirada na competição entre indivíduos pela sobrevivência, porém não mantêm vínculos de dependência ou subordinação.

PG e AGs representam um campo novo de pesquisa dentro da Ciência da Computação. Neste campo muitos problemas continuam em aberto e a espera de novas soluções e ferramentas. Apesar disso, este paradigma vem se mostrando bastante poderoso e muitos trabalhos vêm explorando o uso de AGs e PG para solucionar diversos problemas em diferentes áreas do conhecimento desde mineração de dados e biologia molecular até o projeto de circuitos digitais e inúmeras tarefas envolvendo otimização (GECCO 2000).

## 2. ALGORITMOS GENÉTICOS

O desenvolvimento de simulações computacionais de sistemas genéticos teve início nos anos 50 e 60 através de muitos biólogos, mas foi John Holland que começou a desenvolver as primeiras pesquisas no tema. Em 1975, Holland publicou "*Adaptation in Natural and Artificial Systems*", ponto inicial dos Algoritmos Genéticos (AGs). David E. Goldberg, aluno de Holland, nos anos 80 obteve seu primeiro sucesso em aplicação industrial com AGs. Desde então os AGs são utilizados para solucionar problemas de otimização e aprendizado de máquinas.

Esses algoritmos simulam processos naturais de sobrevivência e reprodução das populações, essenciais em sua evolução. Na natureza, indivíduos de uma mesma população competem entre si, buscando principalmente a sobrevivência, seja através da busca de recursos como alimento, ou visando a reprodução. Os indivíduos mais aptos terão um maior número de descendentes, ao contrário dos indivíduos menos aptos. Os requisitos para a implementação de um AG são:

- Representações das possíveis soluções do problema no formato de um código genético;
- População inicial que contenha diversidade suficiente para permitir ao algoritmo combinar características e produzir novas soluções;
- Existência de um método para medir a qualidade de uma solução potencial;
- Um procedimento de combinação de soluções para gerar novos indivíduos na população;
- Um critério de escolha das soluções que permanecerão na população ou que serão retirados desta;
- Um procedimento para introduzir periodicamente alterações em algumas soluções da população. Desse modo mantém-se a diversidade da população e a possibilidade de se produzir soluções inovadoras para serem avaliadas pelo critério de seleção dos mais aptos.

A idéia básica de funcionamento dos algoritmos genéticos é a de tratar as possíveis soluções do problema como "indivíduos" de uma "população", que irá "evoluir" a cada iteração ou "geração". Para isso é necessário construir um modelo de evolução onde os indivíduos sejam soluções de um problema. A execução do algoritmo pode ser resumida nos seguintes passos:

- Inicialmente escolhe-se uma população inicial, normalmente formada por indivíduos criados aleatoriamente;
- Avalia-se toda a população de indivíduos segundo algum critério, determinado por uma função que avalia a qualidade do indivíduo (função de aptidão ou "fitness");
- Em seguida, através do operador de "seleção", escolhem-se os indivíduos de melhor valor (dado pela função de aptidão) como base para a criação de um novo conjunto de possíveis soluções, chamado de nova "geração";
- Esta nova geração é obtida aplicando-se sobre os indivíduos selecionados operações que misturem suas características (chamadas "genes"), através dos operadores de "cruzamento" ("crossover") e "mutação";
- Estes passos são repetidos até que uma solução aceitável seja encontrada, até que o número predeterminado de passos seja atingido ou até que o algoritmo não consiga mais melhorar a solução já encontrada.

Os principais componentes mostrados na figura 1 são descritos a seguir em mais detalhes.

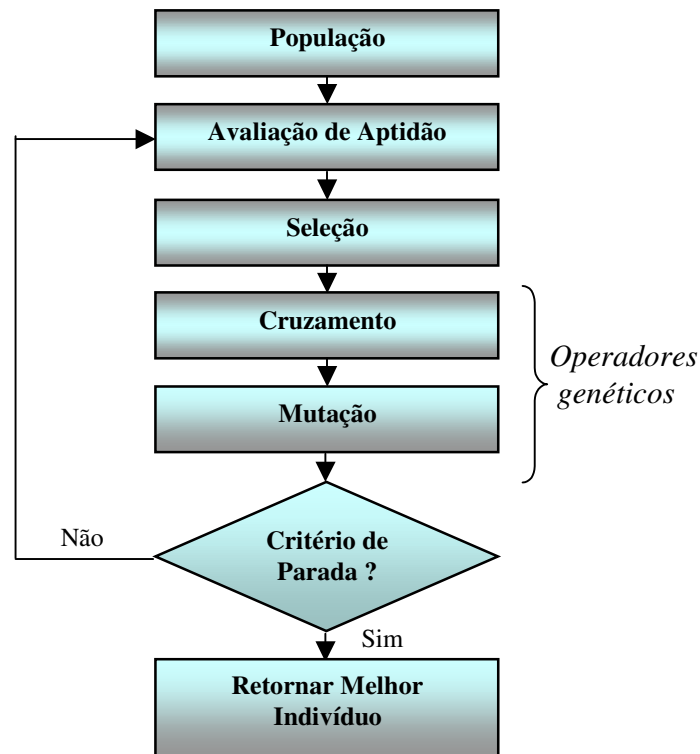


Figura 1 - Estrutura básica de um Algoritmo

## 2.1 População

A população de um algoritmo genético é o conjunto de indivíduos que estão sendo cogitados como solução e que serão usados para criar o novo conjunto de indivíduos para análise. O tamanho da população pode afetar o desempenho global e a eficiência dos algoritmos genéticos. Populações muito pequenas têm grandes chances de perder a diversidade necessária para convergir a uma boa solução, pois fornecem uma pequena cobertura do espaço de busca do problema. Entretanto, se a população tiver muitos indivíduos, o algoritmo poderá perder grande parte de sua eficiência pela demora em avaliar a função de aptidão de todo o conjunto a cada iteração, além de ser necessário trabalhar com maiores recursos computacionais.

## **Indivíduos**

O ponto de partida para a utilização de um algoritmo genético como ferramenta para solução de problemas é a representação destes problemas de maneira que os algoritmos genéticos possam trabalhar adequadamente sobre eles. Uma das principais formas é representar cada atributo como uma sequência de bits e o indivíduo como a concatenação das sequências de bits de todos os seus atributos. Outras variações de codificações binárias podem ser encontradas em (Holland 1975; Caruana 1988).

A codificação usando o próprio alfabeto do atributo que se quer representar (letras, códigos, números reais, etc.) para representar um indivíduo também é muito utilizada. Alguns exemplos podem ser encontrados em (Meyer 1992; Kitano 1994).

Diversas outras formas são possíveis, normalmente a forma mais apropriada está fortemente ligada ao tipo de problema.

### **2.2 Avaliação de Aptidão (Fitness)**

Neste componente será calculado, através de uma determinada função, o valor de aptidão de cada indivíduo da população. Este é o componente mais importante de qualquer algoritmo genético. É através desta função que se mede quão próximo um indivíduo está da solução desejada ou quão boa é esta solução.

É essencial que esta função seja muito representativa e diferencie na proporção correta as más soluções das boas. Se houver pouca precisão na avaliação, uma ótima solução pode ser posta de lado durante a execução do algoritmo, além de gastar mais tempo explorando soluções pouco promissoras.

### **2.3 Seleção**

Dada uma população em que a cada indivíduo foi atribuído um valor de aptidão, existe vários métodos para selecionar os indivíduos sobre os quais serão aplicados os

operadores genéticos. Há diversas formas de seleção, entre eles há o método de seleção por Roleta e o método de seleção por Torneio.

No método de seleção por Roleta (figura 2), cada indivíduo da população é representado na roleta proporcionalmente ao seu índice de aptidão. Assim, para indivíduos com alta aptidão é dada uma porção maior da roleta, enquanto aos indivíduos de aptidão mais baixa, é dada uma porção relativamente menor.

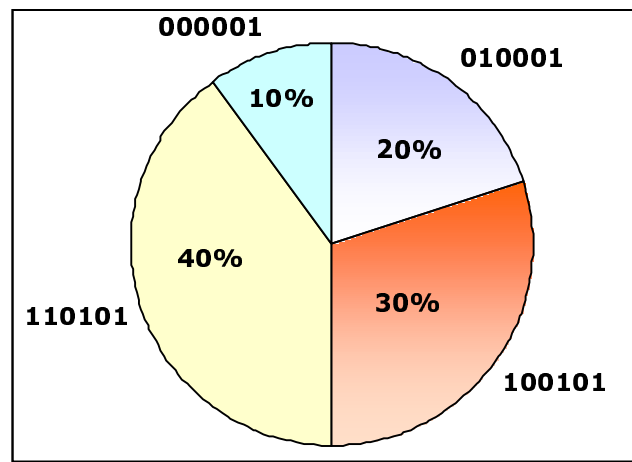


Figura 2 – Método de seleção por Roleta

Neste método, um dos problemas encontrados pode ser o tempo de processamento, já que o método exige duas passagens por todos os indivíduos da população.

Um exemplo da implementação deste método, segundo (Mitchell 1997) é mostrado a seguir na figura 3:

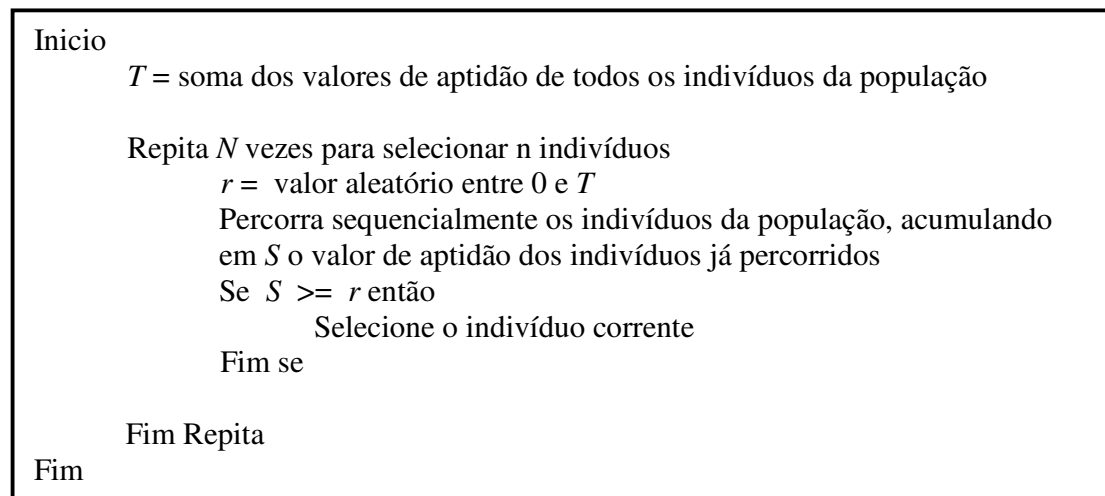


Figura 3 – Algoritmo básico do método de seleção por Roleta

Um outro método é a seleção por Torneio, onde um número  $n$  de indivíduos da população é escolhido aleatoriamente para formar uma sub-população temporária. Deste grupo, o indivíduo selecionado dependerá de uma probabilidade  $k$  definida previamente. Um exemplo básico da implementação deste algoritmo (Mitchell 1997) é mostrado na figura 4, onde  $n=2$ :

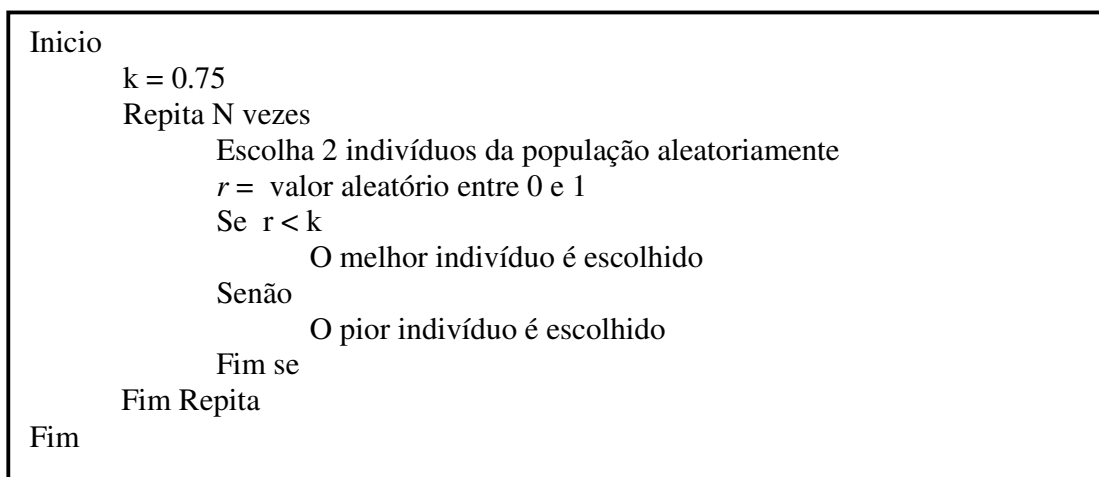


Figura 4 – Algoritmo básico do método de Seleção por Torneio



Este método é o mais utilizado, pois oferece a vantagem de não exigir que a comparação seja feita entre todos os indivíduos da população (Banzhaf 1998).

## 2.4 Operadores Genéticos

O princípio básico dos operadores genéticos é transformar a população através de sucessivas gerações, estendendo a busca até chegar a um resultado satisfatório. Os operadores genéticos são necessários para que a população se diversifique e mantenha características de adaptação adquiridas pelas gerações anteriores. Os operadores de cruzamento e de mutação têm um papel fundamental em um algoritmo genético.

### Cruzamento (Crossover)

Este operador é considerado o operador genético predominante. Através do cruzamento são criados novos indivíduos misturando características de dois indivíduos "pais". Esta mistura é feita tentando imitar (em um alto nível de abstração) a reprodução de genes em células. Trechos das características de um indivíduo são trocados pelo trecho equivalente do outro. O resultado desta operação é um indivíduo que potencialmente combine as melhores características dos indivíduos usados como base.

Alguns tipos de cruzamento bastante utilizados são o cruzamento em um ponto e o cruzamento em dois pontos, mostrados nas Figuras 5 e 6:



Figura 5 – Cruzamento em um ponto

Com um ponto de cruzamento, seleciona-se aleatoriamente um ponto de corte do cromossomo. Cada um dos dois descendentes recebe informação genética de cada um dos pais (Figura 5).

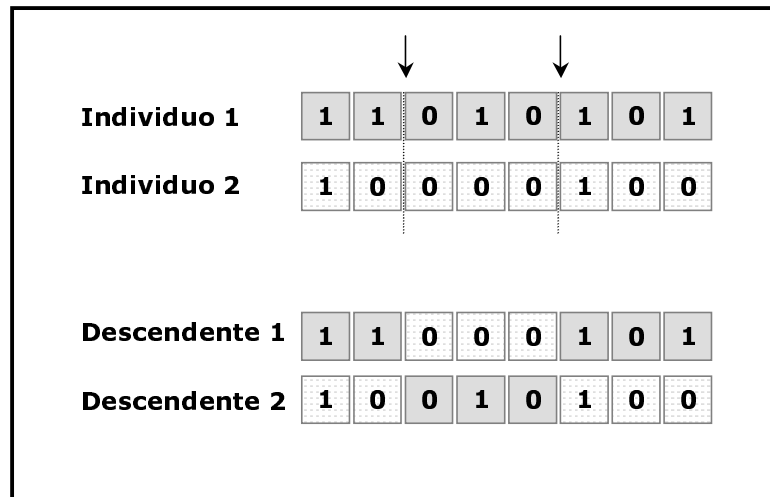


Figura 6 – Cruzamento em dois pontos

Com dois pontos de cruzamento, um dos descendentes fica com a parte central de um dos pais e as partes extremas do outro pai e vice versa (Figura 6).

## Mutação

Esta operação simplesmente modifica aleatoriamente alguma característica do indivíduo sobre o qual é aplicada (ver Figura 7). Esta troca é importante, pois acaba por criar novos valores de características que não existiam ou apareciam em pequena quantidade na população em análise. O operador de mutação é necessário para a introdução e manutenção da diversidade genética da população. Desta forma, a mutação assegura que a probabilidade de se chegar a qualquer ponto do espaço de busca possivelmente não será zero. O operador de mutação é aplicado aos indivíduos através de uma taxa de mutação geralmente pequena.

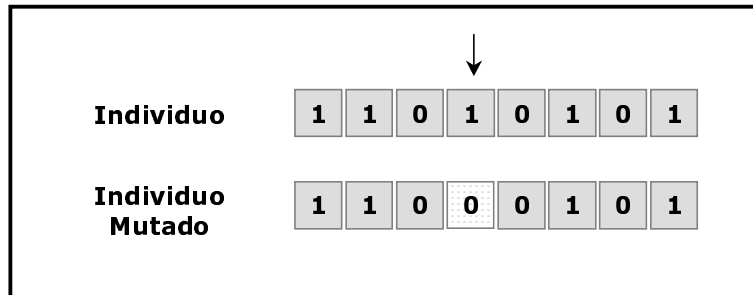


Figura 7 – Mutação Simples

## 2.5 Geração

A cada passo, um novo conjunto de indivíduos é gerado a partir da população anterior. A este novo conjunto dá-se o nome de "Geração". É através da criação de uma grande quantidade de gerações que é possível obter resultados dos Algoritmos Genéticos.

## 2.6 Considerações finais sobre AGs

Os algoritmos genéticos são apropriados para problemas complexos, mas algumas melhorias devem ser feitas no algoritmo básico. Muitas aproximações foram propostas com o objetivo comum de melhorar AGs. O primeiro grupo de estudos foca na manutenção da diversidade na população (De Jong 1989)(Eshelman 1991)(Goldberg 1989)(Goldberg 1990)(Tsutsui1993)(Tsutsui 1994) e inclui: métodos de compartilhamento de recursos que utilizam algumas funções sharing para evitar a convergência de indivíduos semelhantes, métodos crowding que obrigam a substituição de indivíduos novos, restrições de cruzamento, etc.

O segundo grupo visa melhorar o desempenho da capacidade de busca de algoritmos genéticos usando hibridização (Costa1995)(Glover 1994) (Glover 1995) (Kitano 1990)(Malek 1989)(Mantawy 1999)(Muhlenbein 1998)(Muhlenbein 1992)(Powel 1989)(Ulder 1991). Nesta abordagem algoritmos genéticos são usados com um dos seguintes paradigmas: busca tabu, redes neurais artificiais, simulated annealing, etc.

Entretanto, a maioria dos estudos na literatura têm focado na busca global através de AGs, enquanto a busca local tem sido feita por outros métodos.

O último grupo de estudos foca em problemas de funções de otimização, ou em problemas para encontrar soluções ótimas de Pareto (Cantu-Paz 1999)(Coelho 1999)(Schaffer 1985)(Srinivas 1993)(Tamaki 1996)(Fonseca 1993)(Hiroyasu 1999)(Horn 1993). Estes estudos incluem: métodos para dividir indivíduos em subgrupos, cada um representando uma função objetivo, combinação de torneio e métodos de compartilhamento de recursos, métodos para dividir soluções de Pareto em algumas áreas, entre outros.