

3 Computação Evolucionária

Computação Evolucionária compreende diversos algoritmos inspirados no princípio Darwiniano da evolução das espécies e na genética. São algoritmos probabilísticos, que fornecem um mecanismo de busca paralela e adaptativa baseado no princípio de sobrevivência dos mais aptos e na reprodução. A Computação Evolucionária dispensa informação auxiliar e oferece algoritmos gerais (Algoritmos Genéticos, Programação Genética e Algoritmos Evolutivos) que são aplicados em problemas complexos, com grandes espaços de busca, de difícil modelagem, ou para os quais não há um algoritmo eficiente disponível.

Este capítulo apresenta uma breve descrição sobre o princípio de funcionamento dos Algoritmos Genéticos (AG), descrevendo suas partes principais e seus parâmetros de evolução, além de alguns conceitos sobre Algoritmos Genéticos distribuídos.

3.1 Algoritmos Genéticos

Essencialmente, Algoritmos Genéticos são métodos de busca e otimização (Mitchell, 1994; Koza, 1992; Goldberg, 1989; Back, 1996; Fogel, 1966), que têm sua inspiração nos conceitos da teoria de seleção natural das espécies proposta por Darwin.

Os sistemas desenvolvidos a partir deste princípio são utilizados para procurar soluções de problemas complexos ou com espaço de soluções (espaço de busca) muito grande, o que os tornam problemas de difícil modelagem e solução quando se aplicam métodos exatos de otimização.

Estes algoritmos são baseados nos processos genéticos de organismos biológicos para procurar soluções ótimas ou sub-ótimas. Para tanto, procede-se da seguinte maneira: codifica-se cada possível solução de um problema em uma estrutura chamada de "cromossomo", que é composta por uma cadeia de bits ou caracteres. Estes cromossomos representam indivíduos, os quais são então submetidos a um processo evolucionário que envolve avaliação, seleção, cruzamento (*crossover*) e mutação. A cada indivíduo atribui-se um valor de

adaptação, sua aptidão, que indica quanto a solução representada por este indivíduo é boa em relação às outras soluções da população. Após vários ciclos de evolução, a população tende a conter indivíduos mais aptos. Desta maneira, o termo *população* refere-se ao conjunto de todas as soluções com as quais trabalha o algoritmo a cada geração. Aos indivíduos mais adaptados é dada a oportunidade de se reproduzirem mediante cruzamentos com outros indivíduos da população, produzindo descendentes com características de ambas as partes. A mutação também tem um papel significativo, ao introduzir na população novos indivíduos gerados através de perturbações aleatórias realizadas em indivíduos existentes na população, garantindo a diversidade das características dos indivíduos da população.

O processo de evolução começa com a criação aleatória dos indivíduos que formarão a população inicial. A partir de um processo de seleção, baseado na aptidão de cada indivíduo, são escolhidos indivíduos para a fase de reprodução que cria novas soluções, utilizando-se para isto um conjunto de operadores genéticos (basicamente cruzamentos e mutações). Deste modo, a aptidão do indivíduo determina o seu grau de sobrevivência e, assim, a possibilidade que o cromossomo possa fazer parte das gerações seguintes.

O procedimento, básico de um algoritmo genético descrito acima, é resumido na figura 3 (Davis,1991).

```

Início
 $t \leftarrow 1$ 
inicializar população  $P(t)$ 
avaliar população  $P(t)$ 
enquanto (não condição_de_fim) faça
 $t \leftarrow t+1$ 
selecionar população  $P(t)$  a partir de  $P(t-1)$ 
aplicar operadores genéticos
avaliar população  $P(t)$ 
fim enquanto
fim

```

Figura 3. Procedimento Básico do Algoritmo Genético

Para determinar o final da evolução pode-se fixar um número máximo de gerações, ou o número total de indivíduos criados, ou ainda condicionar o algoritmo à obtenção de uma solução satisfatória, isto é, quando atingir um valor

alvo. Outras condições para a parada incluem o tempo de processamento e o grau de similaridade entre os elementos em uma população (convergência).

As seções seguintes apresentam em mais detalhes cada um dos componentes de um algoritmo genético.

3.1.1 Representação

A representação é um aspecto fundamental na modelagem de um AG para a solução de um problema. A solução de um problema pode ser representada por um conjunto de parâmetros (genes), unidos para formar uma cadeia de valores (cromossomo), de maneira que seja capaz de descrever todo o espaço de busca relevante do problema. As soluções (cromossomos) são codificadas através de uma seqüência formada por caracteres de um sistema alfabético. Originalmente, utilizou-se o alfabeto binário (0,1), porém novos modelos de AG's codificam as soluções de forma diferente, como por exemplo números reais (Michalewicz,1996).

Na área de exploração de reservatórios petrolíferos temos, em (Almeida, 2003), um AG para otimização de alternativas de desenvolvimento de reservatórios petrolíferos, onde os cromossomos são codificados com números reais e representam a quantidade, tipo e localização de poços que compõem uma alternativa. Já em Yeten (2003), um AG com representação binária foi utilizado para otimizar a localização, trajetória e número de laterais de um único poço. Nesta tese, tanto a representação binária como real são utilizadas conforme descrito no capítulo 4.

3.1.2 Decodificação

A decodificação do indivíduo (cromossomo) consiste, basicamente, na construção da solução real do problema a partir do cromossomo. O processo de decodificação constrói a solução para que esta seja avaliada pela função de avaliação.

3.1.3 Função de Avaliação

A avaliação é a ligação entre o AG e o problema a ser solucionado. Ela é feita através de uma função que melhor representa o problema e tem por

objetivo oferecer uma medida da aptidão de cada indivíduo na população corrente, que irá dirigir o processo de busca. Dado um indivíduo (cromossomo), a função de avaliação consiste em associar-se um valor numérico, o qual supõe-se proporcional à sua "utilidade" ou "habilidade" do indivíduo representado em solucionar o problema em questão.

Na área de exploração de reservatórios petrolíferos, a função de avaliação mais freqüentemente utilizada é o Valor Presente Líquido (VPL) ou a recuperação acumulada de óleo. Nesta tese, utiliza-se, como função de avaliação, o VPL.

3.1.4 Operadores Genéticos

Os operadores mais conhecidos nos AGs são os de Reprodução, cruzamento (*crossover*) e Mutação.

- **Reprodução:** Refere-se ao processo de selecionar e copiar um determinado cromossomo para a população seguinte, de acordo com sua aptidão. Isto significa que os cromossomos mais aptos (valor de aptidão maior), têm maior probabilidade de contribuir para a formação de um ou mais indivíduos da população seguinte. Existem basicamente os seguintes métodos: troca de toda população; troca de toda população com elitismo, onde todos os cromossomos são substituídos, sendo que apenas o cromossomo mais apto da população corrente é copiado para população seguinte; troca parcial da população (*steady state*), onde os M melhores indivíduos da população corrente são copiados para população seguinte (Koza,1992; Goldberg,1989; Michalewicz,1996).
- **Cruzamento (*crossover*):** É um operador baseado na troca de partes dos cromossomos (pais), formando-se duas novas soluções (filhos). Este processo pode ser observado no exemplo a seguir (Figura 4), onde a solução está codificada com alfabeto binário.

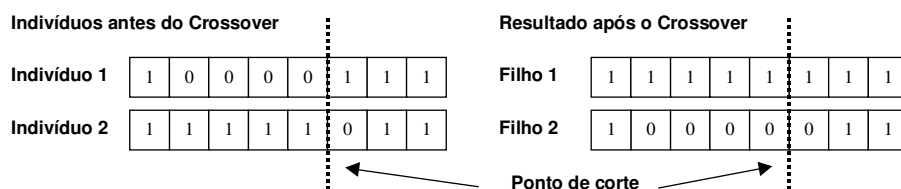


Figura 4. Cruzamento de um Ponto.

O ponto onde ocorre o corte para a realização do cruzamento é escolhido aleatoriamente; no exemplo da figura 4 utilizou-se um único ponto, mas podem ser realizados cortes em mais de um ponto, caracterizando o cruzamento multi-ponto (Goldberg,1989; Michalewicz, 1996; Holland,1992). Para realizar o cruzamento, primeiro é necessária a escolha, por exemplo por sorteio, dos cromossomos “pais”. Em seguida, ocorre a realização ou não do cruzamento segundo um parâmetro, denominado taxa de cruzamento (ver seção 3.1.5). Deste modo, de acordo com a taxa de cruzamento, os cromossomos “pais” podem ser repassados sem modificação para a geração seguinte, criando “filhos” idênticos a eles.

A idéia do operador de cruzamento é tirar vantagem (*exploit*) do material genético presente na população.

- **Mutação:** É a troca aleatória do valor contido nos genes de um cromossomo por outro valor válido do alfabeto. No caso de alfabeto binário, troca-se de 0 para 1 ou vice-versa. Da mesma forma que para o cruzamento, utiliza-se uma taxa de mutação; para cada bit da seqüência de caracteres, sorteia-se se ocorrerá ou não a mutação; no caso de ocorrência, o bit será trocado por outro valor válido pertencente ao alfabeto (Figura 5).

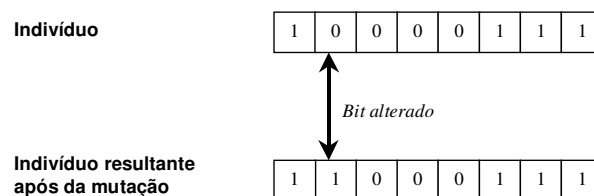


Figura 5. Mutação

A mutação garante a diversidade das características dos indivíduos da população e permite que sejam introduzidas informações que não estiveram presentes em nenhum dos indivíduos. Além disto, proporciona uma busca aleatória (*exploration*) no AG, oferecendo oportunidade para que mais pontos do espaço de busca sejam avaliados.

3.1.5 Parâmetros da Evolução

Os parâmetros que usualmente mais influenciam no desempenho do AG são:

- **Tamanho da População:** o tamanho da população pode afetar o desempenho global e a eficiência dos AG's. Uma população muito pequena oferece uma pequena cobertura do espaço de busca, podendo causar uma queda no desempenho. Uma população suficientemente grande fornece uma melhor cobertura do domínio do problema e tende a prevenir a convergência prematura para soluções locais. Entretanto, com uma população muito grande, tornam-se necessários recursos computacionais maiores, ou um tempo maior de processamento do problema. Logo, deve-se buscar um ponto de equilíbrio no que diz respeito ao tamanho escolhido para a população.
- **Taxa de Cruzamento:** é a taxa que define a probabilidade de um indivíduo ser recombinado com outro. Quanto maior for esta taxa, mais rapidamente novas estruturas serão introduzidas na população. Entretanto, isto pode gerar um efeito indesejável pois a maior parte da população será substituída, ocorrendo assim perda de variedade genética, podendo ocorrer perda de estruturas de alta aptidão e convergência a uma população com indivíduos extremamente parecidos, indivíduos estes de solução boa ou não. Com um valor baixo, o algoritmo pode tornar-se muito lento para oferecer uma resposta aceitável.
- **Taxa de Mutação:** refere-se a probabilidade do conteúdo de um gene do cromossomo ser alterado. A taxa de mutação tende a prevenir que uma dada população fique estagnada em um valor, além de possibilitar que se chegue em qualquer ponto do espaço de busca. Porém, deve-se evitar uma taxa de mutação muito alta, uma vez que pode tornar a busca essencialmente aleatória, prejudicando fortemente a convergência para uma solução ótima.
- **Taxas de Operadores Adaptativas:** Os operadores genéticos podem ser mais úteis no começo ou no final do processo de otimização, dependendo de suas características. Por exemplo, no início do algoritmo genético, a taxa de cruzamento deve ser maior, de forma a aproveitar melhor o material genético inicial; no final da rodada, quando aparecem os efeitos da convergência, deve-se dar ênfase à busca de novo material,

umentando-se a taxa de mutação. Assim, torna-se interessante fazer as taxas dos operadores adaptativas, permitindo variar seus valores durante as gerações (Davis, 1989, 1991).

- **Intervalo de Geração:** controla a porcentagem da população que será substituída durante a próxima geração (substituição total, substituição com elitismo, substituição dos piores indivíduos da população atual, substituição parcial da população sem duplicatas). Esse número de indivíduos substituídos também é conhecido como GAP.
- **Número de gerações:** representa o número total de ciclos de evolução de um Algoritmo Genético, sendo este um dos critérios de parada do AG. Um número de gerações muito pequeno causa uma queda no desempenho, pois não consegue cobrir todo o espaço de busca. Um valor grande acarreta um tempo maior de processamento, mas tende a fornecer uma melhor cobertura do domínio do problema, evitando a convergência para soluções locais.

3.1.6

Avaliação de um algoritmo genético

Uma das formas de avaliar o bom desempenho de um algoritmo genético é através das curvas de evolução: curva *off-line*, curva *online* e *score*.

A curva *off-line* é o valor médio das avaliações dos melhores indivíduos encontrados a cada passo de avaliação até o passo t . Esta curva mostra a qualidade do algoritmo em encontrar soluções boas desde o início.

A curva *on-line* é obtida a partir da média das avaliações de todos os indivíduos avaliados até o passo de avaliação t . Esta curva permite verificar a rápida obtenção de boas soluções e também permite visualizar o grau de convergência dos indivíduos da população.

A curva *score* mostra o melhor indivíduo encontrado em cada geração do algoritmo genético.

3.2

Algoritmos Genéticos Distribuídos

O uso de computação paralela e processos distribuídos é uma tecnologia que vem crescendo nestes últimos anos, dada a crescente demanda por recursos computacionais a baixo custo, pelo aproveitamento da capacidade ociosa em ambientes colaborativos de pesquisa e desenvolvimento, e pela

evolução dos sistemas de rede para sistemas de alto desempenho. O objetivo principal é atingir melhor desempenho partindo do princípio de se juntar processadores, memória e uma rede, a fim de, em conjunto, trabalharem na solução de um dado problema. Nesta seção são apresentados alguns conceitos sobre Algoritmos Genéticos Distribuídos, que vem a ser os próprios algoritmos genéticos inseridos em uma arquitetura de computação paralela. A necessidade de se utilizar tecnologia de computação paralela em AG surge, muitas vezes, devido ao grande número de gerações e indivíduos utilizados no AG, principalmente em problemas complexos, gerando um grande número de avaliações. Além disso, em alguns casos, a avaliação pode ter um custo computacional muito alto.

No exemplo desta tese, a função de avaliação do AG utiliza um simulador de reservatórios, que faz com que a avaliação de cada indivíduo se torne lenta e custosa. Neste caso, é essencial o uso de AG paralelos.

3.2.1 Modelos de Algoritmos Genéticos Distribuídos

Um algoritmo genético, ao ser distribuído em um ambiente de computação paralela, torna-se conveniente pelos seguintes motivos (Túpac, 2005):

- Ganho de tempo de processamento pela distribuição do esforço computacional entre os processadores do ambiente paralelo;
- Vantagens dadas pelo próprio ambiente paralelo pois surgem analogias com a evolução natural existente em populações que estão distribuídas espacialmente.

Um tipo de algoritmo genético distribuído usa os processadores para executar problemas independentes. Isto torna o processo bastante simples em virtude de não haver necessidade de comunicação entre os diferentes processadores; por isso é chamado de *algoritmo paralelo trivial*. Contudo, este método tão elementar possui maior utilidade na execução de várias versões do mesmo problema com diferentes populações iniciais, permitindo gerar estatísticas de desempenho. Dada a natureza estocástica do AG, a obtenção de estatísticas é muito relevante. Outra forma de aproveitar o ambiente paralelo pode ser executando várias versões do mesmo problema com valores diferentes nos parâmetros do AG (taxas de cruzamento e mutação, tamanho da população, número de gerações). Nenhuma dessas metodologias acrescenta novidades na

natureza do AG, mas o tempo de execução pode ser reduzido de forma considerável.

Existem outros modelos de algoritmos genéticos distribuídos, tais como, Algoritmos Genéticos Distribuídos Globais, Algoritmos Genéticos Distribuídos em Ilhas (Cohon, 1987), e Algoritmos Genéticos Distribuídos Celulares (Manderick, 1989 e Tomassini, 1993), nos quais partes específicas da estrutura do algoritmo genético, como a população, a avaliação ou a reprodução, são distribuídas dentro do ambiente paralelo. A seção a seguir concentra-se no modelo de Algoritmos Genéticos Distribuídos Globais, por ser o mais adequado para o desenvolvimento do sistema implementado e utilizado nessa tese.

3.2.2 Algoritmos Genéticos Distribuídos Globais

O algoritmo descrito nessa seção refere-se aos algoritmos genéticos cujo módulo de avaliação é a parte executada dentro do ambiente paralelo. Ainda não surgem grandes mudanças na arquitetura do algoritmo evolucionário, dado que, na maioria dos casos, a avaliação de um indivíduo da população independe das avaliações dos outros indivíduos da mesma população. Contudo, surgem vantagens no sentido de melhor aproveitamento do poder computacional paralelo, dado que, na maioria dos problemas reais, o maior percentual de tempo do algoritmo genético é empregado no cálculo da avaliação. Neste caso, é condição necessária para um melhor aproveitamento do poder paralelo que os tempos de comunicação entre os computadores sejam bem menores do que o tempo necessário para o cálculo da avaliação.

A opção mais imediata é a avaliação simultânea dos indivíduos da população empregando os diferentes processadores da arquitetura paralela. Na arquitetura mestre-escravo Cantú-Paz (1997), um processador mestre executa o ciclo principal do algoritmo genético e, na hora de avaliar uma população de p indivíduos, envia os p indivíduos (cromossomos) para os n processadores escravo disponíveis de forma a serem avaliados. Uma vez obtidos os valores de avaliação, o processador mestre continua com o ciclo do algoritmo genético, aplicando os processos de seleção e evolução para obter a próxima geração de indivíduos.

A situação mais freqüente é o caso em que o número de processadores escravos é menor que o número de indivíduos da população; isto leva à questão de equidade na distribuição das avaliações dos indivíduos nos n processadores.

Isto deve levar em consideração o poder computacional de cada processador no momento da distribuição dos indivíduos. A seguir mostra-se graficamente a arquitetura mestre-escravo.

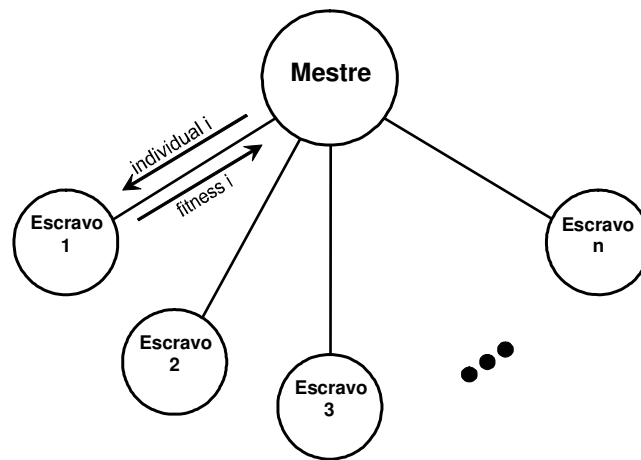


Figura 6. Arquitetura *Master-Slave*

O pseudo algoritmo utilizado neste tipo de distribuição é apresentado a seguir.

```

inicializa a primeira população
para todos os indivíduos executar em paralelo:
  avaliação
fim do executar em paralelo
enquanto não cumprir a condição de finalização
  seleção
  cruzamento
  mutação
  para todos os indivíduos executar em paralelo
    Avaliação
  fim do executar em paralelo
  elitismo
fim do enquanto

```

Figura 7. Pseudo-código: Algoritmo Genético Global

O próximo capítulo descreve o modelo desenvolvido nessa tese.