

3 Métodos de Otimização

Problemas de otimização são relacionados a minimização ou maximização de função de uma ou mais variáveis num determinado domínio, possivelmente com a existência de um conjunto de restrições nessas variáveis. O problema de otimização tratado neste trabalho não é diferente, pois busca-se encontrar um cronograma de abertura de poços que maximize o VPL da alternativa de desenvolvimento e respeite as restrições técnicas e operacionais impostas.

As técnicas de otimização são normalmente utilizadas quando não existe uma solução simples e diretamente calculável para um problema. Isso geralmente ocorre quando a estrutura do problema é complexa, ou existem milhões de soluções possíveis [22].

Um método de otimização é chamado determinístico se for possível prever todos os seus passos até a resposta final. Nesse caso, o método sempre leva à mesma resposta se partir do mesmo ponto. Por isso, a solução encontrada por estes métodos é extremamente dependente do ponto de partida fornecido [23].

Em oposição a estes métodos, existem os chamados métodos estocásticos que, no momento de sua execução, dependem da geração de números aleatórios em algumas de suas etapas. Como a cada execução do algoritmo os números sorteados são diferentes, um método aleatório dificilmente executará a mesma sequência de operações em duas execuções sucessivas, se estiver utilizando sementes de gerador de números aleatórios distintas. Assim, partindo de um mesmo ponto inicial, cada execução do algoritmo seguirá o seu próprio caminho, e possivelmente levará a uma resposta final diferente [23].

De acordo com as características da função objetivo e das restrições pode-se utilizar diferentes métodos para a solução de problemas de otimização. Entre esses métodos destacam-se: o Método de Newton, Método Quase-Newton, Método da Máxima Descida (*Steepest Descent*), Métodos do Gradiente Conjugado, Método das Penalidades, Método do Lagrangeano Aumentado, etc. Todavia, os métodos citados anteriormente podem apresentar domínio de aplicação restrito, pois para

que estes métodos possam ser empregados é necessário que a função objetivo seja contínua e diferenciável no espaço de busca. No entanto, isso não ocorre em muitos dos problemas práticos de engenharia, dificultando assim a sua aplicação. Além disso, quando o número de parâmetros do problema é muito alto, torna-se mais difícil obter o ponto ótimo numericamente [24].

Com a criação de métodos que procuravam reproduzir aspectos do comportamento humano, como adaptação, aprendizado, percepção e raciocínio, surgiu a Computação Natural [25]. A área de Computação Natural pode ser dividida em três grandes sub-áreas: Computação com Mecanismos Naturais, Estudos Sobre Natureza Através da Computação e Computação Inspirada na Natureza [26].

A Computação com Mecanismos Naturais é um novo paradigma de computação, onde os mecanismos naturais, como, por exemplo, as cadeias de DNA, são utilizadas como estrutura de dados para o desenvolvimento de “computadores naturais”. Por conseguinte, os produtos oriundos desta sub-área são vistos como candidatos a complementar ou substituir os computadores digitais disponíveis atualmente. Os principais representantes da Computação com Mecanismos Naturais são: a Computação Molecular e a Computação Quântica [27].

Os Estudos Sobre a Natureza Através da Computação envolvem a utilização de mecanismos computacionais para a síntese de comportamentos naturais, padrões e processos biológicos. As linhas de atuação predominantes são: os Estudos Sobre a Vida e Organismos Artificiais e a Geometria Fractal [27].

A Computação Inspirada na Natureza engloba todas as estratégias desenvolvidas a partir de ou inspiradas em algum mecanismo biológico ou natural. [25] [27]. Ela compreende, principalmente, a Computação Evolucionária, as Redes Neurais Artificiais, a Inteligência Coletiva e os Sistemas Imunológicos Artificiais. Além disso, os próprios aspectos cognitivos e do raciocínio humano representam mecanismos naturais de fundamental importância. Com isso, a Inteligência Artificial Simbólica e os Sistemas Nebulosos, também conhecidos como Sistemas *Fuzzy*, essencialmente relacionados a estes temas, também podem inserir-se no escopo da Computação Natural [24] [25] [27].

As técnicas inspiradas em comportamentos observados na natureza normalmente apresentam caráter aleatório, na tentativa de simular o acaso que

parece governar processos distintos na natureza, desde a evolução das espécies até o comportamento social dos animais. Essas técnicas são comumente denominadas “inteligentes” e são empregadas principalmente em otimização, previsão, controle e inferência. Uma característica de destaque dessas técnicas é a capacidade de oferecer soluções extensíveis a gestores e tomadores de decisões em aplicações complexas, consideradas difíceis e extremamente restritivas [24].

3.1. Computação Evolucionária

Inspirada na genética e no princípio darwiniano da evolução das espécies, a Computação Evolucionária compreende diversos algoritmos probabilísticos que fornecem um mecanismo de busca paralela e adaptativa baseado no princípio da sobrevivência dos mais aptos e na reprodução.

Este mecanismo de busca é obtido a partir de uma população de indivíduos (possíveis soluções) representados por cromossomos (vetores, matrizes ou números binários) associados a aptidões (avaliação da solução do problema). Estes indivíduos são submetidos a um processo de evolução (seleção e reprodução) por vários ciclos [24].

As técnicas de Computação Evolucionária, são úteis em diversos tipos de problemas para o fornecimento de suporte à tomada de decisão, inclusive otimização de cronogramas [28]. Essas técnicas dispensam informações auxiliares como, por exemplo, derivadas das funções que muitas vezes não estão disponíveis ou são difíceis de obter.

3.1.1. Algoritmos Genéticos

Os Algoritmos Genéticos, propostos em [29], são um método de otimização baseado no processo evolutivo natural proposto por Charles R. Darwin (1809-1882). Segundo Darwin: *“Quanto melhor um indivíduo se adaptar ao meio ambiente, maior será sua chance de sobreviver e gerar descendentes”*.

Este algoritmo busca encontrar o indivíduo ótimo de uma população utilizando critérios probabilísticos de seleção e evolução naturais [30][31]. Neste método, codifica-se cada possível solução do problema em uma estrutura chamada cromossomo. Estes cromossomos representam indivíduos, que são evoluídos ao longo de várias gerações, de forma similar aos seres vivos, de acordo com os princípios de seleção natural e sobrevivência dos mais aptos.

Os Algoritmos Genéticos conseguem um equilíbrio entre a capacidade de exploração do espaço de soluções e também de aproveitamento das melhores soluções ao longo da evolução. Com isso, se mostram interessantes para a resolução de problemas complexos de otimização.

Os Algoritmos Genéticos possuem mecanismos intrínsecos que promovem a diversificação de seus indivíduos, além disso, eles são menos suscetíveis ao aprisionamento em ótimos locais. Todavia o tempo de convergência costuma ser mais elevado comparado ao tempo gasto por métodos de otimização baseados em gradiente [29].

Para utilização de alguns métodos baseados em gradientes, como foi mencionado anteriormente, é necessário que a função objetivo seja contínua e diferenciável no espaço de busca. Muitas vezes, isso não ocorre em problema de otimização de cronograma de abertura de poços, quando, por exemplo, os dados de produção são obtidos a partir de um simulador de reservatórios do tipo “caixa-preta”. Este fato dificulta a aplicação de métodos baseados em gradiente, como os métodos de Gauss-Newton e Levenberg-Marquardt [32]. Sendo assim, pode-se acrescentar que os métodos evolutivos são mais adequados para busca de uma solução aceitável, que não necessariamente seja uma solução ótima, para este problema de otimização.

Os Algoritmos Genéticos são tipicamente utilizados para resolver problemas da forma $f: S \rightarrow R$, onde S é um espaço de busca constituído por todas as possíveis soluções para um problema particular. Para codificar possíveis soluções em estruturas de dados do tipo definido no conjunto S , deve ser estabelecida uma representação adequada [33].

Etapas de um Algoritmo Genético Clássico

Dado um problema, o algoritmo genético trata cada possível solução como sendo um indivíduo, e este é representado por um cromossomo, onde as soluções são vetores de números binários. Diz-se então que os cromossomos são compostos por genes (como na biologia) e a cada reprodução estes genes são alterados.

A primeira população de um algoritmo genético clássico, com n indivíduos (potenciais soluções) é criada aleatoriamente. A avaliação desses indivíduos é realizada associando a cada indivíduo da população um número real, também chamado de aptidão, cuja finalidade é medir o quanto aquela situação é adequada para satisfazer a especificação de um problema particular. Uma vez avaliada a população inicial do algoritmo, inicia-se o processo evolutivo, no qual os indivíduos são submetidos às etapas de seleção, cruzamento, mutação e, novamente, avaliação.

Cada indivíduo da população tem uma probabilidade de seleção associada que é diretamente proporcional à sua aptidão. Assim, no processo de seleção os indivíduos passam por um procedimento que, baseado no princípio de Darwin, privilegia os mais aptos. A seleção consiste em distribuir os indivíduos em uma roleta, onde os mais aptos ocupam porções maiores dessa roleta e, ao ser girada, os indivíduos mais aptos terão mais chances de serem selecionados para compor a população da geração seguinte. Como o tamanho da população é mantido constante ao longo de todo o processo evolutivo, na transição de uma geração para a outra, a roleta deve ser girada um número de vezes igual ao tamanho da população. Devido às características da roleta, espera-se que na geração seguinte, a nova população seja formada pelos indivíduos mais aptos da geração anterior e que os menos aptos sejam eliminados. Um exemplo de uma roleta com 5 indivíduos é apresentado na Figura 3.1. Observa-se que o indivíduo identificado como 1 apresenta a maior aptidão, uma vez que ocupa a maior porção. O contrário acontece com o indivíduo identificado como 2. Portanto, a probabilidade de o indivíduo 1 compor a população de indivíduos da geração seguinte é bem maior que a do indivíduo 2. Assim, há uma probabilidade bastante alta de o indivíduo 2 ser eliminado na transição de uma geração para a outra.

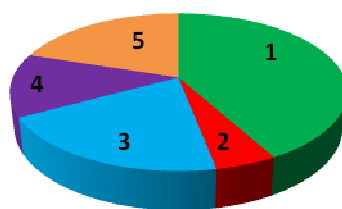


Figura 3.1: Roleta de seleção

Após a seleção, parte dos indivíduos sofre a ação dos operadores genéticos que modificam esses indivíduos, e consequentemente geram novas soluções para o problema em questão. Para isso o algoritmo genético utiliza alguns operadores, como o cruzamento e a mutação.

O cruzamento consiste em recombinar o material genético de dois indivíduos a fim de criar dois novos indivíduos. Com isso, esse operador tem a função de extrair de genes de diferentes indivíduos, e recombiná-los para formar novos indivíduos. O funcionamento do operador de cruzamento pode ser observado na Figura 3.2.

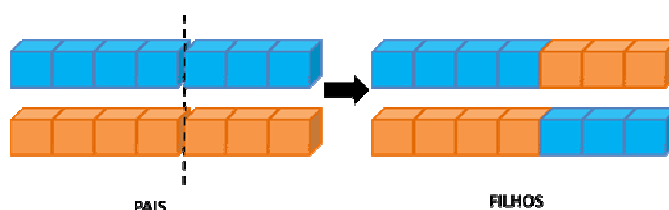


Figura 3.2: Operador clássico de cruzamento

A mutação introduz diversidade em uma população, ou seja, é responsável pela variação dos indivíduos. A mutação consiste em aplicar modificações aleatórias em uma ou mais características de um indivíduo para criar um novo. O funcionamento do operador de cruzamento está ilustrado na Figura 3.3.



Figura 3.3: Operador clássico de mutação

Para a obtenção de bons resultados é fundamental que haja um equilíbrio entre o cruzamento e a mutação, uma vez que o operador de cruzamento é responsável por tirar proveito de um determinado subespaço de busca e o operador de mutação é responsável pela exploração do espaço de busca.

Após a aplicação dos operadores genéticos é originada uma nova população e para encerrar um ciclo de evolução, também chamado de geração, avaliam-se os

indivíduos dessa nova população. Estes ciclos se repetem até que uma condição de parada seja satisfeita. A condição de parada geralmente é dada pelo número de gerações que, juntamente com o tamanho da população e com as taxas de aplicação dos operadores genéticos, compõem o conjunto de parâmetros que devem ser inicialmente informados ao algoritmo genético. Vale destacar que, de um problema para outro, podem ocorrer variações nestes parâmetros. As etapas de um algoritmo genético clássico estão ilustradas na Figura 3.4.

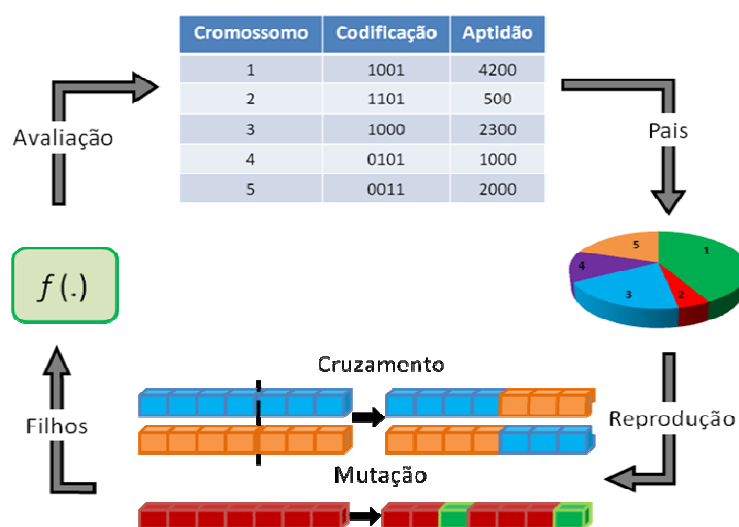


Figura 3.4: Etapas do Algoritmo Genético

Os algoritmos genéticos são capazes de encontrar soluções satisfatórias ou até mesmo a solução ótima, explorando de forma eficiente o espaço de busca. Mesmo quando se tratam de espaços de busca complexos e sujeitos a diversas restrições.

Aprimoramentos dos Algoritmos Genéticos

Uma característica de destaque dos Algoritmos Genéticos é a flexibilidade, uma vez que é possível, a partir da versão clássica, modificar a forma de representação, modificar operadores e até mesmo associar novas técnicas, além de criar modelos híbridos. Essas flexibilidades permitem a criação de modelos de solução mais adequados ao problema a ser resolvido.

A representação por números binários, por exemplo, pode ser substituída por representações baseadas em ordem, números reais, inteiros. Pode haver

também alguma representação específica para o problema, onde esta última pode ser tão complexa quanto as necessidades impostas pelo mesmo. Dentre outras particularidades podem envolver duas ou mais formas de representação no mesmo cromossomo. Para cada tipo de representação é necessário haver operadores de cruzamento e mutação alinhados àquela representação, e dependendo da complexidade da representação é necessário criar operadores específicos.

O procedimento de geração da população inicial pode ser alterado para que nem todos os indivíduos sejam gerados aleatoriamente. Assim, podem-se incluir algumas sementes iniciais, que são indivíduos que representam soluções promissoras previamente conhecidas. Geralmente essas soluções surgem a partir da experiência adquirida por um especialista no problema em questão. Dessa forma, a inclusão de sementes iniciais é atrativa, pois conduz o algoritmo genético a resultados finais melhores comparados a situações nas quais a evolução se inicia com populações geradas aleatoriamente.

Ainda que o melhor indivíduo da população tenha uma probabilidade de seleção maior que a dos indivíduos piores, não há garantia desse indivíduo estar presente na próxima população. Portanto, em determinado momento, essa solução pode ser perdida e isso acarreta uma queda no desempenho da evolução. Isso pode ser contornado através do *steady state*, que garante que um percentual dos melhores indivíduos da população sobreviva para a geração seguinte sem serem submetidos à roleta. Além disso, existe um caso particular do *steady state*, denominado Elitismo, onde apenas o melhor indivíduo sobrevive para a geração seguinte.

O *steady state* pode ser aplicado entre as gerações, como foi explicado, e também entre uma rodada e outra, onde uma rodada representa uma execução completa do algoritmo genético. A realização de várias rodadas é interessante porque permite aumentar a diversidade da população sem perder as melhores soluções encontradas até então.

A utilização de taxas adaptativas também é interessante porque permite que o equilíbrio entre o aproveitamento das melhores soluções e a exploração do espaço de busca seja ajustado ao longo da evolução. Essas possibilidades de modificação tornam os Algoritmos Genéticos extremamente versáteis e, por isso, aplicáveis a problemas de otimização de diversas áreas do conhecimento.