

4

Metáforas de Otimização

O gigantesco avanço tecnológico que vem sofrendo os sistemas de computação, mais precisamente as unidades de processamento, criou a base para o uso efetivo da Inteligência Computacional, permitindo que diversas áreas da ciência possam se beneficiar de uma técnica simples e eficiente na solução de problemas que envolvam otimização e busca.

A Inteligência Computacional é uma ciência que baseia-se em técnicas inspiradas na Natureza no desenvolvimento de sistemas inteligentes, que visam integrar aos sistemas de computação aspectos do comportamento humano, tais como aprendizado, percepção, raciocínio, evolução e adaptação, de forma a buscar soluções para problemas complexos de otimização. Exemplos de três das principais técnicas são: Algoritmos Genéticos (*GA - Genetic Algorithm*), Otimização por Enxame de Partículas (*PSO - Particle Swarm Optimization*) e Recozimento Simulado (*SA - Simulated Annealing*).

Algoritmos Genéticos são algoritmos matemáticos, inspirados nos mecanismos de evolução natural e recombinação genética. A técnica de Algoritmo Genético fornece um mecanismo de busca adaptativa que se baseia no princípio Darwiniano de reprodução e sobrevivência dos mais aptos.

O método Otimização por Enxame de Partículas é um algoritmo de otimização, baseado no comportamento e movimento de animais (peixes, pássaros, etc). Proposto por Eberhart e Kennedy (1995), este método consiste na otimização de uma função objetivo através da troca de informações de estado entre elementos do grupo, resultando em um algoritmo de otimização não determinístico eficiente, robusto e de simples implementação computacional.

O algoritmo Recozimento Simulado explora uma analogia com a termodinâmica ao simular o resfriamento de um conjunto de átomos aquecidos, na busca de estrutura cristalina de energia mínima, operação conhecida como recozimento (*annealing*).

Ambas as técnicas partem de um conjunto de soluções, escolhidas aleatoriamente dentro de um espaço de amostras, que iterativamente são testadas e

evoluem através de um modelo baseado em um processo da natureza.

Os problemas são modelados da seguinte forma: dado um conjunto S de variáveis s (chamadas soluções) e uma função objetivo $f : S \rightarrow \mathfrak{R}$, que associa cada solução $s \in S$ a um valor real $f(s)$, o objetivo é buscar a solução $s^* \in S$, dita ótima ou sub-ótima, para a qual $f(s)$ é mínima. [19]

Em grande parte desses problemas, a modelagem matemática pode levar a funções complexas, não lineares e descontínuas. A minimização dessas funções através de métodos matemáticos convencionais é um problema muito difícil de resolver e por isso são classificados na literatura como NP-difíceis e/ou problemas de otimização combinatória. O uso de métodos exatos para resolução de problemas de otimização combinatória se torna bastante restrito em virtude principalmente da elevada carga computacional exigida. Uma outra questão relevante é que determinados problemas, em virtude da complexidade de encontrar a solução ótima, achar uma solução sub-ótima ao invés do ótimo global é razoavelmente interessante, o qual somente pode ser obtido após um considerável esforço computacional.

Este é o motivo pelos quais os pesquisadores têm concentrado esforços na utilização de metaheurísticas para solucionar problemas desse nível de complexidade. Metaheurísticas podem ser definidas como sendo uma técnica flexível que procura boas soluções (próximas da ótima) a um custo computacional razoável, sem, no entanto, garantir a solução ótima, bem como garantir quão próximo uma determinada solução está da solução ótima. O desafio é produzir, em tempo reduzido, resultados tão próximos quanto possível da solução ótima.

Em suma, esses algoritmos buscam uma solução ótima ou sub-ótima em um determinado domínio, a partir de um conjunto de soluções tentativas que são testadas e comparadas entre si. Em função dessa comparação, as soluções tentativas são submetidas a um processo de evolução, geralmente baseado em uma analogia com algum processo natural. Este processo é orientado a preservar as características das melhores soluções e a descartar as características das soluções piores, produzindo assim um novo conjunto de soluções tentativas que teoricamente estarão mais próximas da solução ótima procurada. O processo evolui até que a melhor das soluções tentativas atinja um determinado patamar de

desempenho ou simplesmente até que um determinado número de evoluções tenha ocorrido.

Evidentemente, esses algoritmos podem não encontrar a solução ótima, ficando em uma solução sub-ótima, principalmente se todas as soluções tentativas convergirem ao longo do processo para essa solução sub-ótima (mínimo local), impedindo assim ou dificultando que a busca continue a explorar todo o domínio (mínimo global). Para evitar a convergência para um mínimo local, o processo evolucionário a que são submetidas as soluções tentativas devem possuir um certo grau de aleatoriedade na geração das novas soluções tentativas, de modo a permitir o surgimento, em pequena escala, de características não presentes nas soluções tentativas originais. Os processos naturais aos quais os algoritmos citados baseiam-se, em geral, também possuem esse pequeno grau de aleatoriedade em sua evolução.

A seguir é feita uma breve descrição de cada uma das metáforas aplicadas no desenvolvimento do trabalho.

4.1

Algoritmo Genético

Algoritmos Genéticos são inspirados no princípio Darwiniano da evolução das espécies e na genética. São algoritmos probabilísticos, inspirados nos mecanismos de evolução natural e recombinação genética, que fornecem um mecanismo de busca paralela e adaptativa baseado no princípio de sobrevivência dos mais aptos e na reprodução, preferencialmente dos melhores indivíduos da espécie.

Os algoritmos genéticos são algoritmos evolucionários que se baseiam no processo natural da evolução das espécies. Biologicamente, toda espécie tem suas características definidas em seus cromossomos. O processo natural da evolução pode ser sucintamente descrito através dos seguintes operados genéticos:

- ▼ Cruzamento (*crossover*) - dois indivíduos de uma mesma espécie cruzam, gerando um novo indivíduo cujos cromossomos serão uma combinação dos cromossomos de seus progenitores. Este novo indivíduo terá, portanto, uma combinação das características de seus antecessores.

Existe um grau de aleatoriedade na forma em que as características serão combinadas;

- ▼ Mutaç o - em uma quantidade pequena de indiv duos, gerados a partir dos cruzamentos, muta es ir o ocorrer, criando caracter sticas nos novos indiv duos que n o existiam em seus antecessores. Novamente, existe um grau de aleatoriedade na determina o de quais cruzamentos ocorrer  uma muta o e qual caracter stica ser  afetada;
- ▼ Sele o - o novo indiv duo, com suas caracter sticas pr prias, ser  submetido ao processo de ‘viver’. Se for apto o suficiente, sobreviver  at  poder tamb m cruzar e passar adiante caracter sticas suas; caso contr rio, perecer .

Dessa forma, ao longo da evolu o da esp cie, o conjunto de indiv duos tende a preservar aquelas caracter sticas que determinam uma maior aptid o. Al m disso, eventualmente, novas caracter sticas ser o introduzidas na esp cie, atrav s da muta o, e a esp cie tende a se tornar cada vez mais apta.

A t cnica de Algoritmo Gen tico produz modifica es na popula o, atrav s da simula o dos mecanismos encontrados na natureza, na evolu o das esp cies, ou seja, essencialmente cruzamento, muta o e sobreviv ncia dos mais aptos. Computacionalmente, os algoritmos gen ticos s o uma abstra o desse processo natural, onde a melhoria da esp cie, tornando-se cada vez mais apta,   a busca de uma solu o sub- tima para um problema, onde o conjunto de caracter sticas dispon veis aos indiv duos s o o espa o de busca do problema e cada indiv duo representa uma poss vel solu o para o problema.

O desenvolvimento de um esquema de otimiza o que usa o fundamento proposto pelo Algoritmo Gen tico pode ser descrito nas seguintes fases:

- ▼ Identifica o do Problema;
- ▼ Representa o das poss veis solu es;
- ▼ Inicializa o;
- ▼ Avalia o;
- ▼ Sele o;

▼ Operadores Genéticos.

4.1.1

Identificação

No contexto do presente trabalho, a identificação do problema consiste na definição da função a ser minimizada denominada função objetivo. Esta função é a descrita na seção 3.1.

4.1.2

Representação

A representação das possíveis soluções do espaço de busca de um problema, ou seja, a representação dos indivíduos (também chamados de cromossomos) define a estrutura do cromossomo a ser manipulado pelo algoritmo. A representação do cromossomo depende exclusivamente do tipo de problema tratado. [20]

Nos primórdios do GA, a representação das soluções de problemas numéricos era através de *strings* binárias, cuja representação é simples, fácil de manipular e de aplicar os operadores genéticos.

No presente trabalho, a representação dos cromossomos é feita através de um vetor de dimensão n (onde n é igual ao número de variáveis da função objetivo), composto de números inteiros não negativos. Cada variável, no caso específico do GA, é um gene (um cromossomo é composto de genes).

4.1.3

Inicialização

Consiste na definição dos indivíduos que compõem a população inicial (ou seja, dos cromossomos iniciais), que iniciarão o processo de evolução e busca da solução sub-ótima.

Informações quanto a região de contorno que possa ser encontrada a solução do problema, auxilia a convergência do algoritmo. Este tipo de informação, em alguns casos, pode ser obtida a partir de uma análise prévia na função objetivo.

4.1.4

Avaliação

A avaliação do problema tem por objetivo fornecer uma medida de aptidão de cada cromossomo na população corrente. A medida de aptidão influencia diretamente na seleção dos “melhores” indivíduos para geração dos descendentes que irá dirigir o processo de evolução e busca.

4.1.5

Seleção

O processo de seleção em algoritmos genéticos seleciona indivíduos para a reprodução. A seleção é baseada na aptidão dos indivíduos: os indivíduos mais aptos têm maior probabilidade de serem escolhidos para reprodução, e define como probabilidades são atribuídas a cada cromossomo.

No contexto deste trabalho, como o mesmo trata de minimizar uma função objetivo, a idéia mais sensata é associar um valor de probabilidade de seleção de cada cromossomo que seja inversamente proporcional a aptidão do mesmo. Assim, se $f(s_i)$ é a avaliação do indivíduo s_i na população corrente, a probabilidade p_{s_i} do indivíduo s_i ser selecionado é proporcional a

$p_{s_i} = (f(s_i))^{-1} / \sum_{n=1}^N (f(s_n))^{-1}$. A idéia é aplicar a média ponderada com peso $1/f(s_i)$.

Uma vez definida a probabilidade, os cromossomos que serão utilizados para gerar a descendência são escolhidos aleatoriamente, através do método de roleta. Os cromossomos são ordenados conforme probabilidade associada, formando uma roleta. Um número é sorteado no intervalo $[0,1]$. O cromossomo associado a fatia da roleta do número gerado aleatoriamente é escolhido como um dos reprodutores para a próxima geração de indivíduos, o qual será aplicado os operados apresentados a seguir.

4.1.6

Operadores Genéticos

Indivíduos selecionados (e reproduzidos na população seguinte) são recombinaados sexualmente, através do operador de *crossover*, ou então são “modificados” com a introdução de um material genético novo, através da mutação.

A metodologia de operação dos cromossomos está diretamente relacionada com a maneira como estes são representados. Operar geneticamente uma string binária é uma operação trivial e consiste de um “embaralhamento” entre strings (no caso do *crossover*) ou na troca de um ou mais bits 0 por 1 e vice-versa (no caso da mutação).

Abstraindo pequenas variações com relação as características herdadas pelos descendentes quando do *crossover* [21], pode-se estender esta operação no contexto de representação deste trabalho.

Uma implementação complementar a solução convencional de *crossover*, apresentada em [21], amplia o espaço de busca do algoritmo. Como se trata de representação através de números inteiros não negativos, a solução convencional escolhe aleatoriamente os genes que serão recombinaados (*crossover* uniforme). A implementação complementar (*crossover* linear) funciona da seguinte forma: seja os cromossomos pais s_1 e s_2 , três novos cromossomos são gerados a partir das seguintes operações: $(\frac{1}{2}s_1 + \frac{1}{2}s_2)$, $(\frac{3}{2}s_1 - \frac{1}{2}s_2)$ e $(\frac{3}{2}s_2 - \frac{1}{2}s_1)$. Destes três, os dois cromossomos que apresentarem valores menores da função objetivo, sobreviverão para a geração seguinte.

Com relação a mutação, a solução implementada define limiares superior e inferior no intervalo de mutação do gene a ser modificado.

O fluxograma abaixo sumariza o funcionamento do Algoritmo Genético.

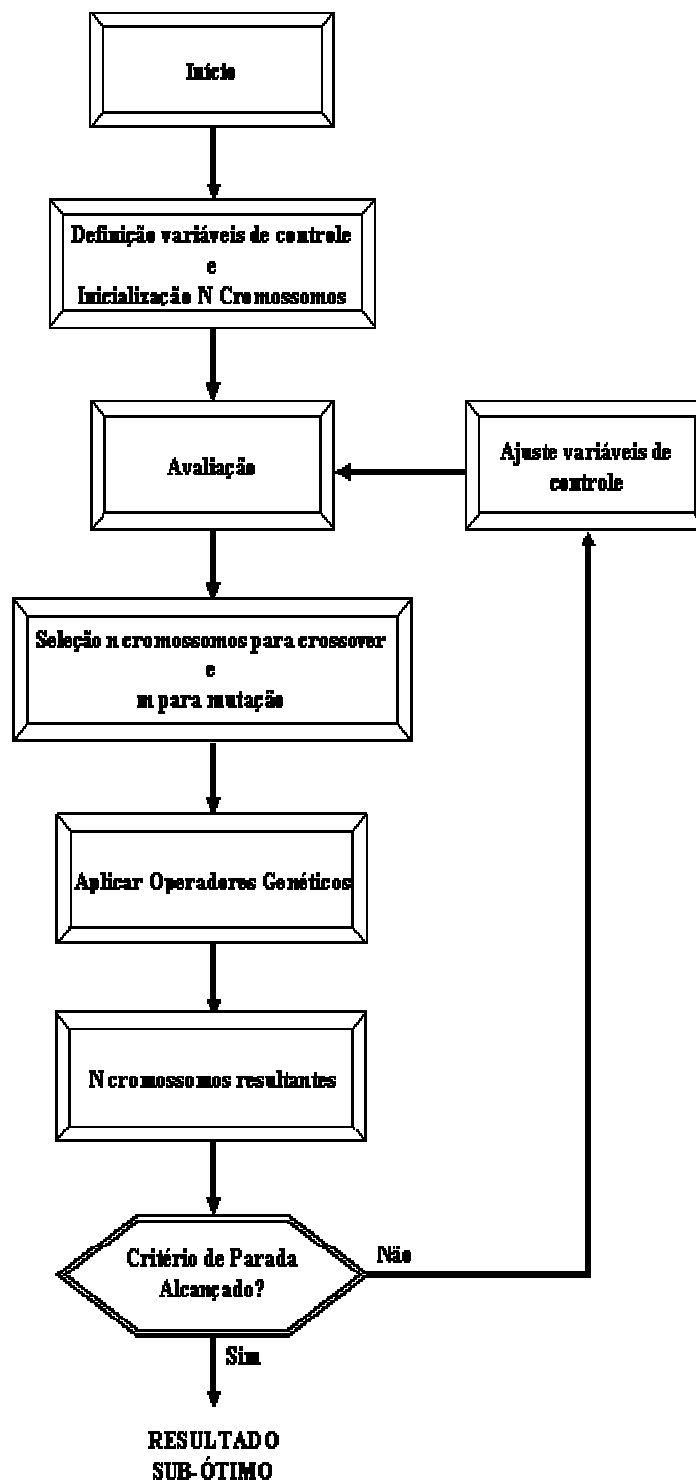


Figura 4.1 – Fluxograma Algoritmo Genético

4.2

Otimização por Enxame de Partículas (*PSO - Particle Swarm Optimization*)

PSO é uma técnica de computação evolucionária, proposta por Eberhart e Kennedy, baseada na observação do comportamento social de um grupo de indivíduos na natureza. PSO trabalha com um grupo de posições de indivíduos e produz modificações na posição desses indivíduos, baseado no comportamento dos indivíduos dentro de um grupo social. O comportamento de cada indivíduo é baseado na sua experiência anterior e na experiência daqueles outros indivíduos com os quais se relaciona.

Similarmente aos algoritmos genéticos, o conjunto de indivíduos tende a preservar aquelas posições que determinam uma maior aptidão e a descartar as posições de menor aptidão. Computacionalmente, os algoritmos ‘swarm’ são uma abstração desse processo natural, onde a procura pela posição mais apta é a busca de uma solução ótima para um problema, onde o conjunto de possíveis posições dos indivíduos é o espaço de busca do problema, e cada posição ocupada por um indivíduo representa uma possível solução para o problema. [24]

Computacionalmente, a implementação de algoritmos PSO para a otimização de um problema pode ser feita através dos seguintes passos:

1º Passo) Gera-se uma população inicial, onde a posição de cada indivíduo representa uma solução possível para o problema. A escolha inicial é feita de forma aleatória em um espaço de busca para o problema, e a posição é avaliada por uma função de aptidão. Gera-se também uma matriz de deslocamento para cada indivíduo, também de forma aleatória, de forma que todos os indivíduos se desloam de suas posições iniciais;

2º Passo) A nova posição de cada indivíduo é definida a partir da matriz de deslocamento. Cada indivíduo tem seu deslocamento ajustado em função da aptidão de suas posições anteriores e da aptidão dos indivíduos do grupo. Cada indivíduo se desloca de acordo com o seu ajuste de deslocamento. E assim sucessivamente.

As equações a seguir modelam o movimento de cada indivíduo [25]:

$$\blacktriangledown \quad v_{\text{Novo}} = v_{\text{Antigo}} + c_1 \times \text{rand} \times (p_{\text{best}} - p_{\text{corrente}}) + c_2 \times \text{rand} \times (g_{\text{best}} - \text{corrente});$$

$$\blacktriangledown \quad p_{\text{corrente}} = \text{corrente} + v_{\text{Novo}} .$$

onde:

v_{Antigo} = parâmetro que define a velocidade corrente de cada indivíduo;

v_{Novo} = parâmetro que define a velocidade que, a cada iteração, ajusta a posição de cada indivíduo (observar que o ajuste na posição leva em consideração a melhor aptidão entre todos os indivíduos de todas as iterações, definidos pelos parâmetros p_{best} e g_{best});

p_{best} = parâmetro que armazena a melhor posição alcançada por cada indivíduo;

g_{best} = parâmetro que armazena a melhor posição alcançada entre todos os indivíduos do grupo;

rand = número aleatório no intervalo [0,1];

$c_1 = c_2 = 2$ (normalmente).

4.3

Recozimento Simulado (*Simulated Annealing*)

O algoritmo Recozimento Simulado (SA - *Simulated Annealing*), foi originalmente proposto por Kirkpatrick, em 1983 e por Cerny, em 1988. Considera o processo de fundição (*annealing*) de um sólido. Consiste de um método de busca local que aceita soluções piores, durante o processo de busca, como mecanismo de escape de ótimos locais.

O SA simula um método natural, fundamentado numa analogia com a termodinâmica, ao simular o modo como um metal se resfria e congela numa estrutura cristalina de energia mínima, operação conhecida como Recozimento [22]. O termo e operação de Recozimento são amplamente utilizados na metalurgia.

O processo de Recozimento, em poucas palavras, parte de “temperaturas elevadas” e, através de uma redução progressiva e suave da temperatura, busca o estado cristalino de energia mínima do sólido. Este processo pode ser visto como a busca pela melhor solução.

O SA, assim como os algoritmos descritos anteriormente, é um algoritmo usado para problemas de otimização, onde a função objetivo corresponde à energia dos estados de um sólido.

Inicialmente, o algoritmo SA requer a definição de um ponto central, a partir do qual será feito o processo de busca. Em cada fase deste processo, o ponto central corrente possui uma região de contorno (vizinhança) que será percorrida na procura da solução sub-ótima. Esta região é ajustada dinamicamente através de parâmetros de controle que influenciam o processo de arrefecimento.

O parâmetro de controle que ajusta esta região é a “Temperatura” (T_{Temp}). A Temperatura permite distinguir entre alterações profundas ou ligeiras na função objetivo. Alterações drásticas ocorrem a elevadas temperaturas e modificações pequenas a temperaturas baixas. No início do processo de busca, a temperatura normalmente é elevada. Como consequência, a região de contorno possui dimensões consideráveis, o que possibilita uma busca mais grossa.

Um outro parâmetro denominado “Taxa de Resfriamento” (T_{ARREF}), controla o processo de arrefecimento. Este parâmetro é responsável por suavizar a supressão da região de contorno, o que refina o processo de busca.

Cada temperatura define uma probabilidade de aceitação de soluções de maior energia (soluções piores) durante a excursão pela região de contorno. A probabilidade de aceitar soluções piores evita o problema de ficar preso num ótimo local. Entretanto, uma característica importante é que a probabilidade de se aceitar um vizinho de maior energia decresce com o tempo, a medida que a temperatura decresce. A equação que define esta probabilidade é apresentada abaixo:

$$P(T) = \frac{1}{e^{\frac{\Delta f}{T}}}$$

Por fim, em qualquer temperatura, dados dois vizinhos de maior energia que o candidato a mínimo corrente, A e B, se $energia(A) > energia(B)$, a probabilidade

de aceitação de A será menor que a de B.

O fluxograma a seguir sumariza o funcionamento do SA.

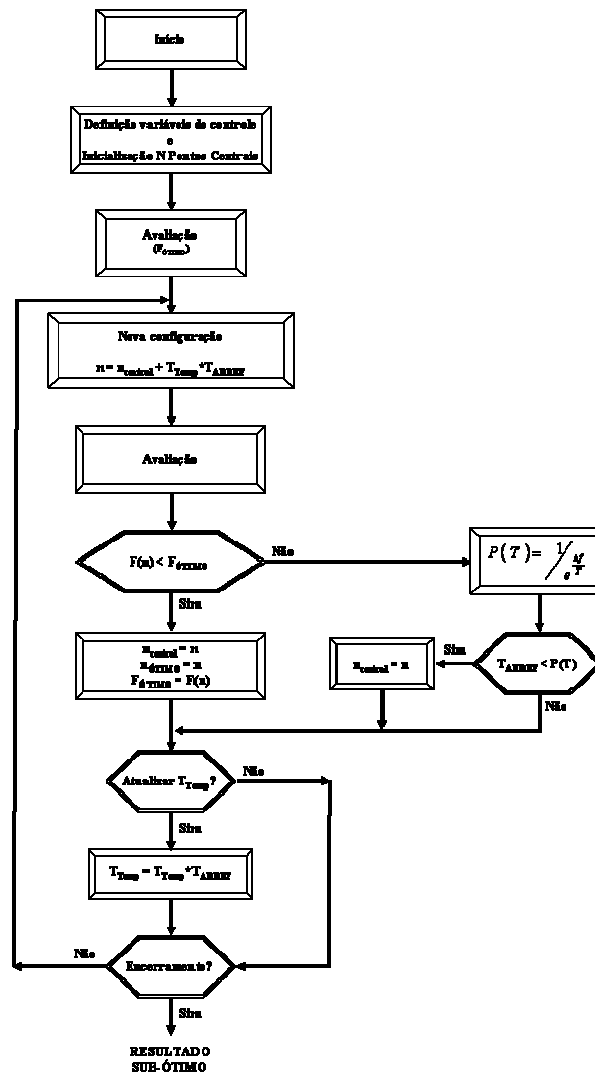


Figura 4.2 – Fluxograma *Simulated Annealing*