

Algoritmos Genéticos

Aplicações em Otimização de funções

Unicamp, 2018

- O desenvolvimento de Algoritmos Genéticos foi influenciado por trabalhos de pessoas, como: Lamark, Malthus, Darwin e Mendel.
- Tornou-se popular após a introdução de John Holland em 1975 no livro *Adaptation in Natural and Artificial Systems*.

Por que usar Algoritmos Genéticos?

- Para ilustrar como um algoritmo genético pode ser útil iremos considerar o problema de uma macaco tentando escrever “to be or not to be that is the question.”

Por que usar Algoritmos Genéticos?

- Para ilustrar como um algoritmo genético pode ser útil iremos considerar o problema de uma macaco tentando escrever “to be or not to be that is the question.”
- Em um dicionário com 26 letras e um espaço, a probabilidade é de um em 27^{39} .

Por que usar Algoritmos Genéticos?

- Para ilustrar como um algoritmo genético pode ser útil iremos considerar o problema de uma macaco tentando escrever “to be or not to be that is the question.”
- Em um dicionário com 26 letras e um espaço, a probabilidade é de um em 27^{39} .
- Mas e se ele pudesse começar com 200 frases aleatórias e evoluir a partir destas utilizando alguma método de avaliar quão boa é cada frase?

Componentes de um Algoritmo Genético

Seleção Natural de Darwin

- **Hereditariedade** - Deve haver um processo pelo qual uma criança receba características de seus pais.
- **Variedade** - Deve haver uma variedade de características e/ou algum meio pelo qual variação seja introduzida. Sem variação toda criança será idêntica a seus pais e nenhuma evolução ocorrerá.
- **Seleção** - Deve haver algum mecanismo pelo qual membros de uma população sejam selecionados a passar suas características a seus filhos. Além disso, a probabilidade de um membro mais bem adaptado ao seu contexto de ter um filho deve ser maior do que o de um membro menos adaptado.

Componentes de um Algoritmo Genético

Implementação

- **População** - Uma população inicial deve ser gerada com uma certa quantidade N de membros, cada um possuindo características aleatórias.
- Deve se decidir previamente quais as características que desejamos evoluir. Por exemplo, se a população é formada por criaturas tentando sobreviver, então provavelmente desejamos evoluir a capacidade de coletar comida.

Componentes de um Algoritmo Genético

Implementação

- **Seleção** - A seleção é dada em duas partes, na primeira avaliamos o quão bom cada membro é através de uma função de adequação, na segunda criamos um conjunto contendo os membros que estão aptos para procriar.
- É importante salientar que a probabilidade de um membro reproduzir deve estar relacionada com sua adaptação.

Componentes de um Algoritmo Genético

Implementação

- **Reprodução** - Criado o conjunto de membros aptos a reproduzir selecionamos aleatoriamente dois membros para cruzar e gerar um novo membro que fará parte da próxima geração. Repetimos o processo até termos N membros na nova população.
- O cruzamento entre dois membros é feito por uma função Cruzamento que passa a criança características de ambos pais.
- A criança deve ainda poder sofrer mutação, ou seja, apresentar características que não estavam presentes em nenhum de seus pais.

Função de adequação

- Função que desejamos maximizar.
- Os genes de cada criatura devem influenciar o valor da função, apesar de que não é estritamente necessário que sejam entradas da função.
- No exemplo anterior poderíamos atribuir como função de adequação quanto tempo a criatura vive.
- Em otimização podemos nos referir a função de adequação como função objetivo.

Conjunto de Reprodutores

- O conjunto é formado por todos membros da população que possam reproduzir, é associado ainda a cada membro uma probabilidade.
- Há diversas maneiras de se criar este conjunto, no caso do exemplo anterior, o conjunto é formado por todas criaturas vivas.
- Outras maneiras serão apresentadas com posteriores exemplos.

Representando os genes

- Os genes são representados por vetores, cada posição representa uma característica, por exemplo digamos que cada criatura tenha genes dado por (Força, Velocidade), então cada membro possui uma quantificação destes, como (10, 3) ou (2, 9).
- Em alguns casos é importante ainda diferenciar genótipo de fenótipo.

Genótipo x Fenótipo

- **Genótipo** - guardam a informação dos genes como visto anteriormente.
- **Fenótipo** - responsáveis pela forma como os genes são expressos.
- As criaturas, fenótipos, são a forma como os genes se expressam no mundo.

Função Cruzamento

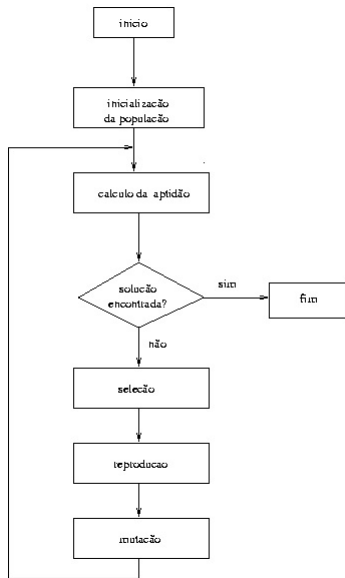
- Esta função é responsável pela forma como os genes dos pais serão transmitidos ao filho.
- A forma mais comum é decidir uma posição r no vetor de genes e passar a criança os genes de um até r do pai e o resto da mãe.
- Se fizermos $r = 1$ para nossas criaturas, então a criança recebe a Força do pai e Velocidade da mãe.

- Durante a reprodução há sempre uma probabilidade de que a criança sofra uma mutação em algum dos genes.
- Esta probabilidade é chamada de **Taxa de Mutação** e previamente determinada.
- Digamos que a chance de uma de nossas criaturas sofrer mutação é de 0.01, então em 1% das reproduções teremos crianças apresentando Força e/ou Velocidade diferente de ambos pais.
- Isto nos garante que soluções diferentes serão buscadas, evitando a estagnação em um mínimo ou máximo local.

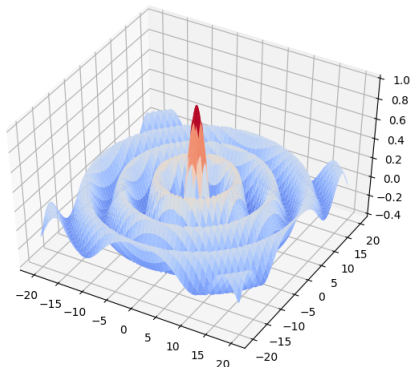
Algoritmo Genético

- 1 **Inicialização:**
 - Cria uma população de N criaturas, cada uma com genes gerados aleatoriamente.
- 2 **Seleção:**
 - Avalie a função de adequação de cada criatura e crie um conjunto de reprodução.
- 3 **Reprodução:** Repita N vezes:
 - Selecione dois pais com probabilidade de acordo com a adequação.
 - Cruzamento: crie uma criança ao combinar os genes destes pais.
 - Mutação: mutacione os genes da criança.
 - Adicione a criança a nova população.
- 4 Substitua a antiga população pela nova e retorne ao passo 2.

Diagramação



Otimização de Funções



- Vamos maximizar a função de Bessel de ordem 0.
- O ponto ótimo é $(x,y) = (0,0)$ com valor 1.0.

Exemplo: Parte I - Inicialização

- Os genes serão dados pelo vetor de coordenadas (x, y) .
- Utilizaremos uma população de 1000 membros.
- Nossa taxa de mutação será 0.05.
- Precisamos ainda de limitantes para o problema, ou seja, valores máximos e mínimos para x e y . Utilizaremos 6 como limitante superior e -6 como inferior para ambas coordenadas.
- Falta ainda decidir: Como criaremos o conjunto de reprodução, a função Cruzamento e como a mutação ocorrerá.

Exemplo: Parte II - Conjunto de Reprodução

- Utilizaremos um método denominado por **Seleção por Torneio**.
- Comece com conjunto vazio M. Para cada membro da população faça:
 - 1 Selecione aleatoriamente um outro, e diferente, membro.
 - 2 Adicione a M aquele com maior valor na função de adequação.

Exemplo: Parte III - Função Cruzamento

- Implementaremos uma função Cruzamento extremamente simples.
- Dados um pai e uma mãe, passe ao filho o valor de x pertencente ao pai e o valor de y pertencente a mãe.

Exemplo: Parte III - Mutação

- Dada a taxa de mutação tM .
- Seja v um número gerado aleatoriamente entre 0 e 1, se $v < tM$ então passe ao filho novos genes completamente aleatórios.

Exemplo: Parte IV - Critério de Parada

- Número máximo de gerações sem melhorar a solução.
- Solução de todos membros da população são parecidas.

$$\begin{aligned} & \text{Max } x + y \\ \text{s.a : } & 5x + 3y \leq 15 \\ & 6x + 4y \leq 24 \\ & x \geq 0, y \geq 0 \end{aligned} \tag{1}$$

- Solução $(x,y) = (\frac{10}{3}, 1)$ com valor 4.33.

Otimização de Funções restritas

- Vamos realizar três mudanças em relação ao código anterior.

Otimização de Funções restritas

- Vamos realizar três mudanças em relação ao código anterior.
- A função de adequação ainda será a função objetivo, porém, para cada restrição não satisfeita dividimos a adequação por 10.

Otimização de Funções restritas

- Vamos realizar três mudanças em relação ao código anterior.
- A função de adequação ainda será a função objetivo, porém, para cada restrição não satisfeita dividimos a adequação por 10.
- A função de mutação realiza quatro etapas:
 - 1 Gere número aleatório r entre 0 e 1.
 - 2 Se $r < tM/20$: gere novos genes.
 - 3 Se $r < tM/10$: perturbe os genes em um raio R_0 .
 - 4 Se $r < tM$: perturbe os genes em um raio $R > R_0$.

Otimização de Funções restritas

- Vamos realizar três mudanças em relação ao código anterior.
- A função de adequação ainda será a função objetivo, porém, para cada restrição não satisfeita dividimos a adequação por 10.
- A função de mutação realiza quatro etapas:
 - 1 Gere número aleatório r entre 0 e 1.
 - 2 Se $r < tM/20$: gere novos genes.
 - 3 Se $r < tM/10$: perturbe os genes em um raio R_0 .
 - 4 Se $r < tM$: perturbe os genes em um raio $R > R_0$.
- O melhor membro da geração anterior está sempre presente na nova geração.

Vantagens e Desvantagens

Desvantagens

- Necessitamos de uma região escolhida previamente.
- A quantidade de parâmetros que devem ser escolhidos previamente é relativamente alta.

Vantagens

- Flexibilidade para ser aplicado em diversos problemas.
- Flexibilidade de customização.
- Obtenção de bons resultados práticos.