

Fundamentos de Inteligência Artificial [5COP099]



Dr. Sylvio Barbon Junior

Departamento de Computação - UEL

Disciplina Anual

Aula 18

Algoritmos Genéticos

Sumário

- Introdução
- Problemas de Busca
- Conceitos Biológicos
- Algoritmo Genético Tradicional
 - Representação da População
 - Estratégia de Seleção
 - Operadores de Busca
 - Cruzamentos
 - Mutação
- Convergência
- Caixeiro Viajante

Introdução

- **Algoritmos Genéticos** (GA - Genetic Algorithms) são uma família dos modelos de computação inspirados pela teoria da evolução (Computação Evolutiva), que são:
 - Estratégia Evolutiva: Ênfase na auto-adaptação;
 - Programação Genética: Indivíduos são armazenados em árvores sintáticas;
 - Programação Evolutiva: Baseado somente em seleção e Mutação.

Introdução

- Os algoritmos são baseados na codificação de potenciais soluções de problemas em estruturas que emulam os cromossomos.
- A ideia é aplicar **recombinações** dos **indivíduos** mais **aptos** de uma **população** para oferecer a melhor solução para um problema de busca e **otimização**.
 - **Problema de Busca**: visa encontrar a melhor (ou viável) solução no espaço do problema (todas as possíveis soluções). Em parte dos casos a busca é também uma otimização.
- Inspirados na teoria da evolução de *Charles Darwin*.

Problemas de Busca

- Em um problema de busca, a função objetivo avalia cada solução com "uma nota", investigando uma solução que corresponda ao ponto de máximo (ou mínimo) da função objetivo.
- Exemplos de técnicas de busca:
 - **Busca Aleatória:** Os pontos no espaço de busca são selecionados aleatoriamente e sua aptidão (fitness) calculados.
 - **Gradiente:** Também chamado de subida de montanha (Hill Climbing), tem a busca conduzida pela derivada da função a ser otimizada. A função deve ser derivável e unimodal (se for multimodal apresenta problemas de máximo local).
 - **Busca Iterativa:** Trata-se da combinação de Busca Aleatória com Gradiente.
 - **Recozimento Simulado:** (simulated annealing), apresenta modificação do método Gradiente para evitar máximos (ou mínimos)

Problemas de Busca

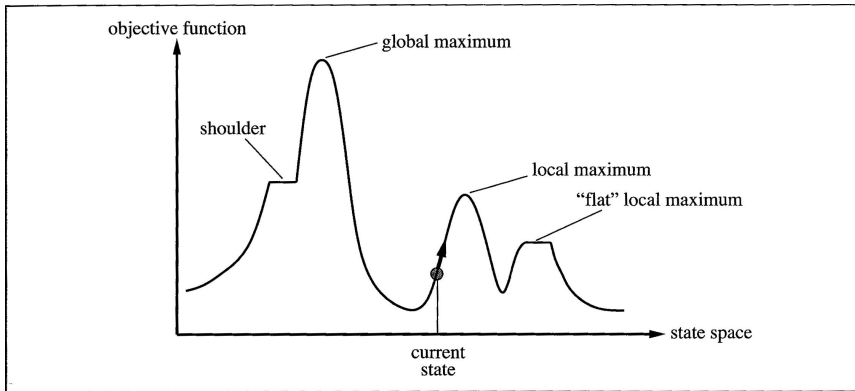


Figure 4.1 A one-dimensional state-space landscape in which elevation corresponds to the objective function. The aim is to find the global maximum. Hill-climbing search modifies the current state to try to improve it, as shown by the arrow. The various topographic features are defined in the text.

Problemas de Busca - Dificuldades

- Os problemas de busca reais podem ter espaços de busca muito grandes.
- Muitos algoritmos não são capazes de localizar o ótimo global na presença de múltiplos ótimos locais (ex. Hill Climbing).
- Os algoritmos genéticos são uma boa solução pois introduzem o conceito de viés particular para cada problema, onde a cada nova geração, espera-se tratar somente indivíduos de uma população seriam já adaptados.
- No entanto, os algoritmos genéticos são métodos conhecidos da Inteligência Artificial como "weak methods", pois assumem que a cada geração o problema está sendo resolvido.
- Os AGs, mesmo sendo Weak Methods, são robustos e gerais para os mais variados problemas. Atualmente trabalham em

Conceitos Biológicos

- **Cromossomos:** String de Genes que são modelos para cada organismo.
- **Gene:** Codifica uma proteína (ou característica) que define o organismo.
- **Locus:** Local no cromossomo ocupado pelo gene.
- **Genoma:** Conjunto de todos os cromossomos.
- **Genótipo:** Conjunto particular de genes.
- **Indivíduos:** São os organismos de uma determinada população.

Conceitos Biológicos

- **Competição:** É a conquista por recursos, envolvendo também a conquista de um par para perpetuação da espécie.
- **Aptidão:** Indivíduos aptos a sobreviver e atrair um par, geram uma descendência maior. É medida pela sua capacidade de sobreviver ao ambiente.
- **Recombinação:** (Crossover) É a combinação dos genes dos cromossomos para formar um novo cromossomo.
- **Mutação:** É o evento que pode ocorrer durante a mutação. Seu efeito é a mudança de um gene.

AG - Tradicional

- Para os algoritmos genéticos, o indivíduo mais apto é a solução para o problema avaliado.
- A população são as possíveis soluções para o problema.
- Para cada indivíduo é calculado um grau de aptidão (Fitness Score).
- Os indivíduos com os maiores Fitness Score se reproduzem, aplicando sobre os genes descendentes mutações.
- O processo é iterativo até que seja atingido o critério de parada.

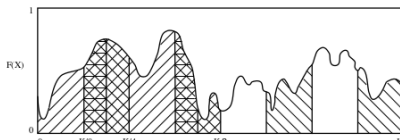
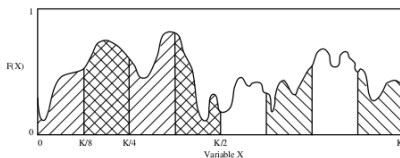
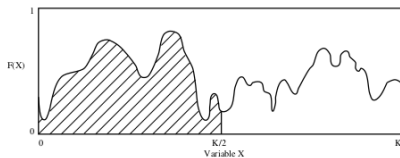
Aula 18 - Algoritmos Genéticos

AG - Tradicional

- Questões importantes:
 - Representação dos Indivíduos;
 - Estratégia de Seleção;
 - Operadores de Busca;

Aula 18 - Algoritmos Genéticos

Problemas de Busca



AG - Representação de Indivíduos

- Assume-se que um indivíduo pode ser representado por um conjunto de parâmetros;
- Os parâmetros são unidos para formar uma String (cromossomo);
- As representações tradicionais são:
 - **Codificação Binária:** É a mais simples, cada cromossomo é uma série de bits (0 ou 1).
 - **Codificação por Permutação (Path):** cada cromossomo é uma série de números que representa uma posição sequencial.
 - **Codificação por Valores:** cada cromossomo é uma sequência de valores como por exemplo letras ou variáveis nominais.
 - **Codificação Real:** Muito utilizada em problemas de otimização, apresentam valores reais como gene.

AG - Representação de Indivíduos

- **Codificação Binária:**

- Permite muitas combinações, porém não é natural para muitos problemas, muitas vezes exigindo codificações e recombinações.
- É a representação mais tradicional e fácil de manipular.
- Exemplo de cromossomos:
 - Cromossomo A : 101100101100101011100101
 - Cromossomo B : 111111100000110000011111

- **Codificação por Permutação:**

- É útil para problemas de ordenação, porém exige correções para manter as recombinações e correções consistentes e reais dentro do universo do problema.
- Exemplo de cromossomos:
 - Cromossomo A : 1 5 3 2 6 4 7 9 8
 - Cromossomo B : 8 5 6 7 2 3 1 4 9

AG - Representação de Indivíduos

- **Codificação por Valores:**

- Cada cromossomo pode assumir valores como caracteres ou qualquer outro.
- Pode exigir um complexo método para recombinação e mutação específico para o problema.
- Exemplo de cromossomos:
 - Cromossomo A_1 : ABDJEIFJDHDIERJFDLDFLFEGTC
 - Cromossomo X_1 : (atrás), (atrás), (direita), (frente), (esquerda)

- **Codificação Real:**

- Usa parâmetros contínuos e uma representação binária é muito custosa ou inadequada. Os parâmetros numéricos podem ser diretamente codificados nos cromossomos.
- Exemplo de cromossomos:
 - Cromossomo A : 1.2324 5.3243 0.4556 2.3293 2.4545
 - Cromossomo B : 7.8924 2.3113 2.4966 1.1193 3.4115

AG - Estratégia de Seleção

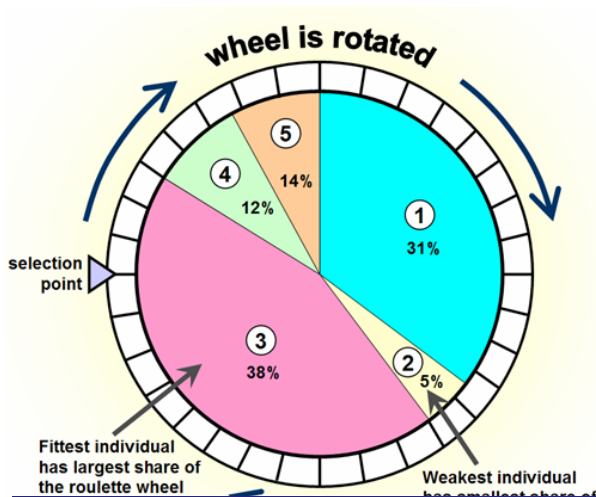
- A seleção tem a função de selecionar um indivíduo (ou par de indivíduos) para geração de descendentes.
- Os indivíduos mais aptos de uma população geram novas "soluções»
- Em geral, indivíduos pais são selecionados com uma probabilidade proporcional a seus valores de fitness.
- Aptidão de um indivíduo é calculada pelo **Fitness Function** que deve retornar um valor único.
- A função de Aptidão é justamente a descrição do problema que se deseja solucionar.

AG - Estratégia de Seleção

- Existem vários mecanismos de seleção, entre eles:
 - **Roleta** (*roulette*): Ordena-se as aptidões da população, calcula-se as aptidões acumuladas, gera-se um número aleatório. O indivíduo selecionado é o primeiro com aptidão acumulada maior que o número aleatório gerado. Roleta tem problemas quando há grandes diferenças entre os valores de aptidão.
 - **Torneio** (*tournament*): Escolhe-se inicialmente n indivíduos, seleciona-se o de maior aptidão, itera-se o processo até preencher o critério esperado.

Aula 18 - Algoritmos Genéticos

AG - Estratégia de Seleção



AG - Operadores de Busca

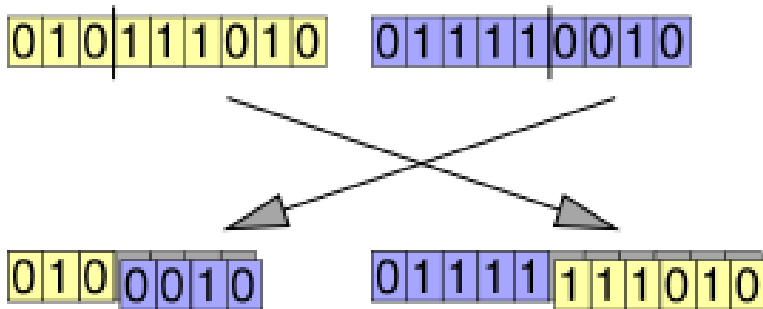
- Também chamados como operadores genéticos, geram uma população intermediária de potenciais cromossomos pais;
- A produção de filhos é realizada até completar o tamanho da população desejada.
- Os operadores mais comuns são:
 - **Recombinação ou Cruzamento (crossover):** é o operador que simula a troca de material genético entre os ancestrais;
 - **Mutação:** Tem a função de aumentar a variabilidade da população. Ocorre com pequena probabilidade.

AG - Operadores de Busca

- **Recombinação ou Cruzamento:**

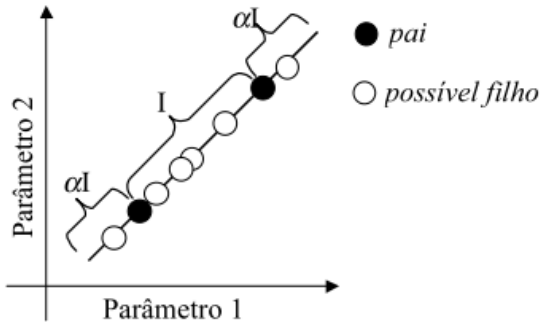
- Nem sempre é aplicada, probabilidade entre 0,6 e 1,0.
- Se não é realizada, os pais são replicados na nova geração.
- Na sua forma mais simples, pais tem seus cromossomos cortados aleatoriamente e recombinados.
- Apresenta diversas formas, entre elas:
 - **Single-point crossover:** um único ponto aleatório de corte;
 - **Multi-point crossover:** vários pontos aleatórios de corte;
 - **Uniform crossover:** dada uma máscara, os genes são copiados
 - **Média Aritmética:** o filho é obtido com $(Pai_1 + Pai_2) / 2$
 - **Média Geométrica:** o filho é obtido com $\sqrt{Pai_1 \times Pai_2}$
 - **Operador BLX - α :** o filho é obtido com $Pai_1 + \beta \times (Pai_2 - Pai_1)$, sendo $\beta = [-\alpha, 1 + \alpha]$

AG - Cruzamentos - Single-point



Aula 18 - Algoritmos Genéticos

AG -Cruzamentos - *BLX* - α



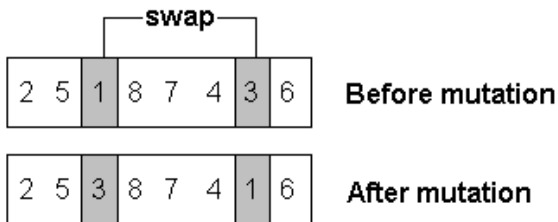
AG - Operadores de Busca

- **Mutação**

- É aplicada a cada descendente após a recombinação;
- Ela modifica aleatoriamente cada gene, ocorrendo com uma pequena probabilidade (0,001).
- No seu caso mais simples, para uma codificação binária, basta inverter o bit.
- Tradicionalmente deve ter menos influência na evolução do que a recombinação.
- Exemplos de Mutação:
 - **Reciprocal Exchange**: Dois genes são sorteados, e os valores trocam de posição;
 - **Inversion**: Um segmento é selecionado e sua ordem é trocada;
 - **Creep**: Um pequeno valor acrescenta ou subtrai sobre o gene selecionado aleatoriamente;
 - **Gaussiana**: Um gene é aleatoriamente selecionado e seu valor é substituído usando uma distribuição Normal

Aula 18 - Algoritmos Genéticos

AG - Mutação - Reciprocal Exchange

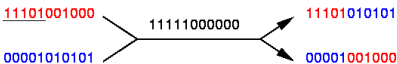


Aula 18 - Algoritmos Genéticos

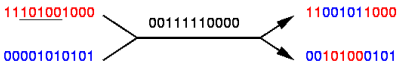
AG - Mutação - Point Mutation

Initial strings *Crossover Mask* *Offspring*

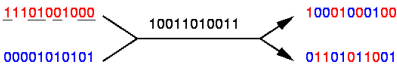
Single-point crossover:



Two-point crossover:



Uniform crossover:



AG - Convergência

- Se o AG foi implementado corretamente, a aptidão da população em geral deve crescer até um valor ótimo global.
- Entende-se por convergência uma progressão até a uniformidade;
- Diz-se que um gene convergiu quando 95% da população compartilha do mesmo valor;
- Alguns critérios de parada podem ser estabelecidos:
 - Número máximo de iterações;
 - Função Objetivo com valor máximo alcançado;
 - Convergência da função objetivo (quando não ocorre melhoria significativa);

AG - Convergência

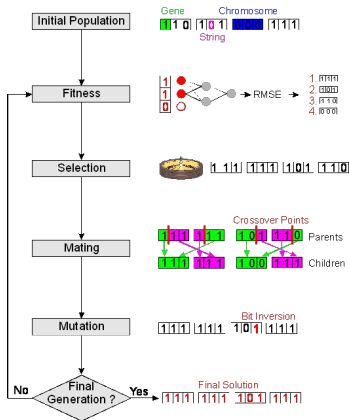
- A População inicial não deve ser muito pequena, pois não irá representar o problema ou conduzir a máximos locais;
- Caso a População seja excessivamente grande, a convergência demorará;
- O ideal é representar a população uniformemente amostrada no espaço de busca;
- A maioria dos projetos que usam AG ajustam regras empíricas para melhorar o seu desempenho;
- Não existe uma regra restrita para explicar sua forma por completo.

AG - Evitando Convergência Prematura

- Alguns cenários e modelagens podem levar à convergência prematura e incorreta. Para evitar estes problemas, podemos:
 - **Escala de Aptidão** (Fitness Scaling): Define-se um valor de oportunidade de reprodução, subtraindo uma constante da aptidão pela média de aptidões modificadas. Aumenta a razão entre a aptidão máxima e a média.
 - **Janelamento de Aptidão** (Fitness Windowing): Subtrai-se o valor da aptidão mínima nas últimas gerações e divide-se pela médias das aptidões modificadas.
 - **Ordenamento de Aptidão** (Fitness Ranking): Ordena-se os indivíduos de acordo com a Aptidão, normaliza-se as aptidões entre a máxima e a média.

Aula 18 - Algoritmos Genéticos

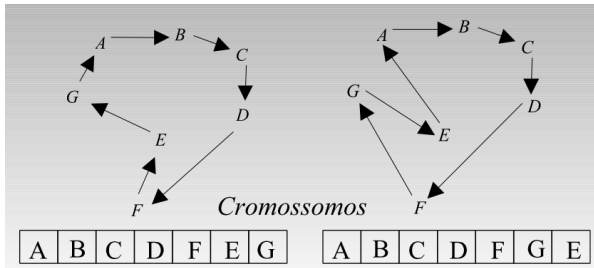
AG - Resumo



Aula 18 - Algoritmos Genéticos

Exemplo - Caixeiro Viajante

- Dado um número de cidades, encontrar o caminho mais curto passando por todas as cidades uma única vez.
- A função objetivo é a distância total percorrida.



Exemplo - Caixeiro Viajante

- Cruzamento baseado em posição:

<i>pai1</i>	A	B	C	D	F	E	G
<i>pai2</i>	C	E	G	A	D	F	B
<i>filho1</i>	B	E	C	A	D	F	G
<i>filho2</i>	C	B	E	D	F	G	A

Aula 18 - Algoritmos Genéticos

Exemplo - Caixeiro Viajante

- Mutação baseada em troca de ordem:



Lista de Referências

1. Henneth A. de Jong, *Evolutionary Computation: A Unified Approach*, MIT, 2006
2. M. Dorigo, *Optimization, Learning and Natural Algorithms*, PhD thesis, Politecnico di Milano, Italie, 1992
3. Kennedy, J.; Eberhart, R. (1995). "Particle Swarm Optimization". *Proceedings of IEEE International Conference on Neural Networks*. IV. pp. 1942–1948. doi:10.1109/ICNN.1995.488968.
4. Shi, Y.; Eberhart, R.C. (1998). "A modified particle swarm optimizer". *Proceedings of IEEE International Conference on Evolutionary Computation*. pp. 69–73.
5. Kennedy, J.; Eberhart, R.C. (2001). *Swarm Intelligence*. Morgan Kaufmann. ISBN 1-55860-595-9.