

Algoritmos Genéticos

Prof. Marcelo Ricardo Stemmer

marcelo@das.ufsc.br

Introdução

- Algoritmos genéticos (AG) são parte da **Computação Evolucionária**, que é uma área em crescimento da Inteligência Artificial.
- A Computação Evolucionária foi introduzida em 1960 por **Rechenberg** com seu trabalho "*Estratégias de Evolução*" (*Evolutionsstrategien* no original).
- AG originado de estudos sobre autômatos celulares de **John Holland**, da Universidade de Michigan, publicados em 1975 no livro "*Adaption in Natural and Artificial Systems*".
- AG inspirados na teoria da evolução de **Charles Darwin** (sobrevivência do mais apto), apresentada na obra "*A Origem das Espécies*" publicada em 1859.

Introdução

- Algoritmos Genéticos são métodos adaptativos que podem ser usados para resolver problemas de **busca e otimização**.
- Dado um problema, o conjunto de todas as possíveis soluções é chamado de **espaço problema** (“problem space”) ou **espaço de busca** (“search space”).
- Um algoritmo de busca é um procedimento para encontrar a melhor (ou a mais útil) solução no espaço problema (ou espaço de busca).
- Na maioria dos casos a busca é também uma otimização.

Introdução

- Outras técnicas de busca:
 - **Busca Aleatória:** pontos no espaço de busca são selecionados de forma aleatória e sua aptidão é calculada;
 - **Método do Gradiente** (subida de montanha): busca guiada pela derivada da função a ser otimizada, que tem que ser derivável e unimodal (se função multimodal => máxima local e não global);
 - **Busca Iterativa:** combinação dos anteriores;
 - **Recozimento Simulado** (simulated annealing): modificação do método do gradiente para evitar máximos (ou mínimos) locais.

Bases biológicas

- **Cromossomos** são strings de **genes** que servem de modelo para o organismo inteiro.
- cada **gene** codifica uma proteína em particular, que define uma característica do organismo (por exemplo, a cor dos olhos).
- cada gene ocupa um lugar determinado no cromossomo, chamado **locus**.
- conjunto completo de cromossomos é chamado de **genoma**.
- um conjunto particular de genes forma um **genótipo**.
- o **genótipo** determina certas características do organismo após o nascimento, definindo o **fenótipo** (estatura, cor dos olhos, inteligência, etc.).

Bases biológicas

- Na natureza, **indivíduos** de uma população **competem** por **recursos** como alimento, água e abrigo.
- A **competição** envolve em geral também a conquista de um **par** para reprodução (**perpetuação da espécie**).
- Indivíduos mais **aptos** a sobreviver e atrair um par geram em geral uma **descendência maior**, isto é, o genótipo destes indivíduos se difunde mais que os outros.

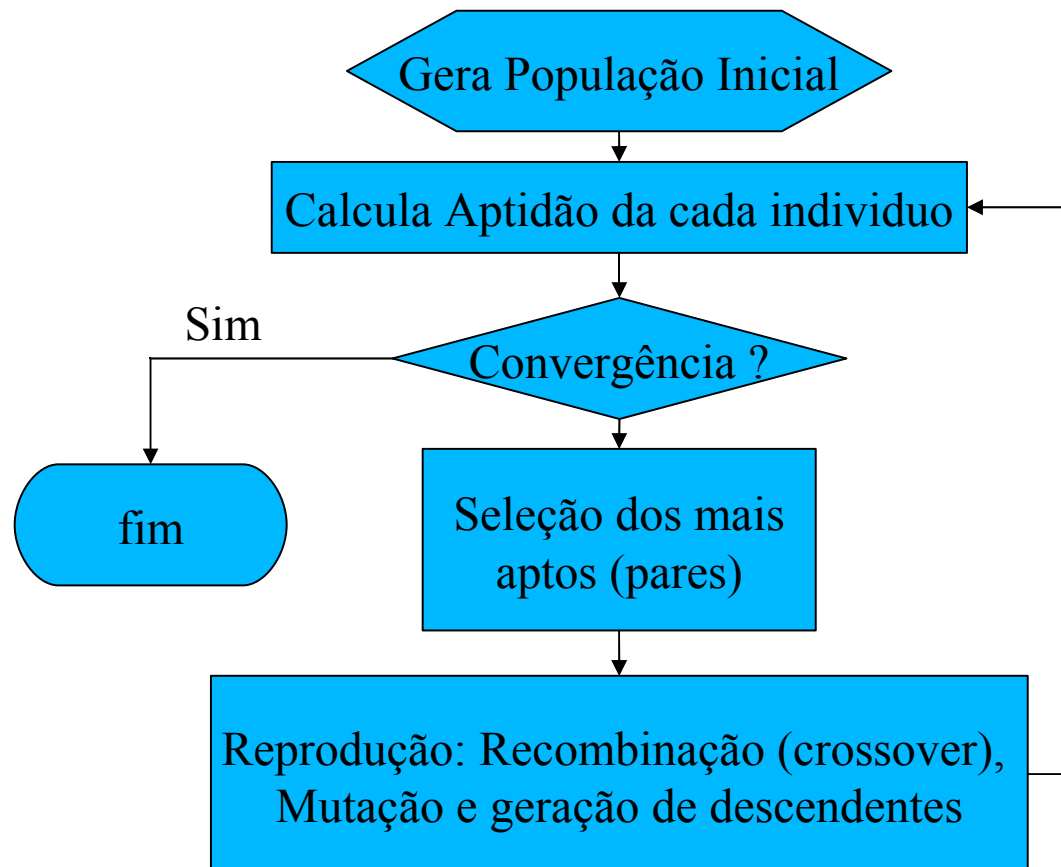
Bases biológicas

- Durante a reprodução, há uma **recombinação** (“crossover”) de genes, na qual os genes dos pais se combinam para formar um novo cromossomo.
- O novo cromossomo pode sofrer **mutação**, que implica em mudanças aleatórias nos genes (devidas a fatores ambientais, como radiação, ou a simples erros de cópia).
- A **recombinação** e a **mutação** dos genes de indivíduos aptos pode gerar descendentes **super-aptos** (mais aptos do que os ascendentes) => Evolução!
- A **aptidão** (“fitness”) do organismo é medida pela sua capacidade de sobreviver no ambiente.

AG básico (canônico)

- Em AG, o “**indivíduo**” que evolui é uma **solução** para um dado problema.
- Este “indivíduo” faz parte de uma “**população**”, composta de possíveis soluções ao problema.
- Cada indivíduo (solução) recebe uma **pontuação de aptidão** (“fitness score”), que indica quão boa é a solução para o problema.
- Soluções com “fitness score” mais alto se reproduzem, sendo que os genes do descendentes sofrem recombinação e mutação.
- Processo se repete até algum critério de parada ser atingido.

AG básico



Geração da População: Codificação

- Assume-se que uma solução possa ser representada por um conjunto de parâmetros (genes).
- Estes parâmetros são unidos para formar um string (cromossomo).
- Holland sugeriu que melhor representação do cromossomo é um alfabeto binário.
- Os parâmetros que formam o cromossomo são o **genótipo**. O organismo resultante é o **fenótipo**.

Codificação

- Na **Codificação Binária** (mais usada e mais simples), cada cromossomo é uma série de **bits** - **0** ou **1**.

Cromossomo A

101100101100101011100101

Cromossomo B

111111100000110000011111

Codificação

- Codificação Binária permite muitos possíveis cromossomos, mesmo com pequenos número de alelos.
- Por outro lado, esta codificação não é natural para muitos problemas e algumas vezes é necessário fazer correções antes das recombinações e/ou mutações.
- **Exemplo de Problema:** Problema da Mochila
O problema: É dada uma lista de coisas com preços e tamanhos. É fornecido o valor da capacidade da mochila. Escolha as coisas de forma a maximizar o valor daquilo que cabe dentro da mochila, sem ultrapassar sua capacidade.
Codificação: Cada bit é usado para dizer se a coisa correspondente está ou não na mochila.

Codificação

- Na **Codificação por Permutação**, cada cromossomo é uma série de números que representa uma posição em uma **seqüência**.

Cromossomo A

1 5 3 2 6 4 7 9 8

Cromossomo B

8 5 6 7 2 3 1 4 9

Codificação

- A Codificação por Permutação é útil para solução de problemas de ordenação.
- Para alguns tipos de recombinações e mutações, são necessárias correções para que os cromossomos fiquem consistentes (isto é contenham seqüências reais) para alguns problemas.

- **Exemplo de Problema:** Problema do Caixeiro Viajante (Travelling Salesman Problem - TSP)

O problema: São dadas cidades e as distâncias entre elas. O caixeiro viajante tem que visitar todas elas, sem viajar mais do que o necessário. A solução do problema consiste em encontrar a seqüência de cidades em que as viagens devem ser feitas de forma que a distância percorrida seja a mínima possível.

Codificação: Os cromossomos descrevem a ordem em que o caixeiro visitará as cidades.

Codificação

- Na **Codificação de Valores**, cada cromossomo é uma seqüência de alguns valores. Esses valores podem ser qualquer coisa relacionada com o problema, tais como: números reais, caracteres ou qualquer outro objeto.

Cromossomo A

1.2324 5.3243 0.4556 2.3293 2.4545

Cromossomo B

ABDJEIFJDHDIERJFDLDFLFEGTC

Cromossomo C

(atrás), (atrás), (direita), (frente), (esquerda)

Codificação

- Codificação de Valores é uma boa escolha para alguns problemas especiais.
- Entretanto, para essa codificação, é frequentemente necessário desenvolver um método de recombinação e mutação específico para o problema.
- **Exemplo de Problema:** Cálculo de pesos para uma rede neural (treinamento)

O problema: É dada uma rede neural com arquitetura definida. Encontre os pesos entre os neurônios da rede de forma a obter a resposta desejada da rede.

Codificação: Valores reais dos cromossomos representam os pesos da rede neural.

Cálculo da Aptidão

- A **aptidão (fitness)** de um individuo depende do desempenho do fenótipo.
- A aptidão é calculada a partir do genótipo através de uma “**Função de Aptidão**” (**Fitness Function**), que deve retornar um valor numérico único (figura de mérito).
- A Função de Aptidão depende do problema a resolver.
- Ela pode ser óbvia em alguns casos: se o objetivo é encontrar o ponto de máxima ou mínima de uma função, a função de aptidão pode ser a própria função a otimizar.

Seleção

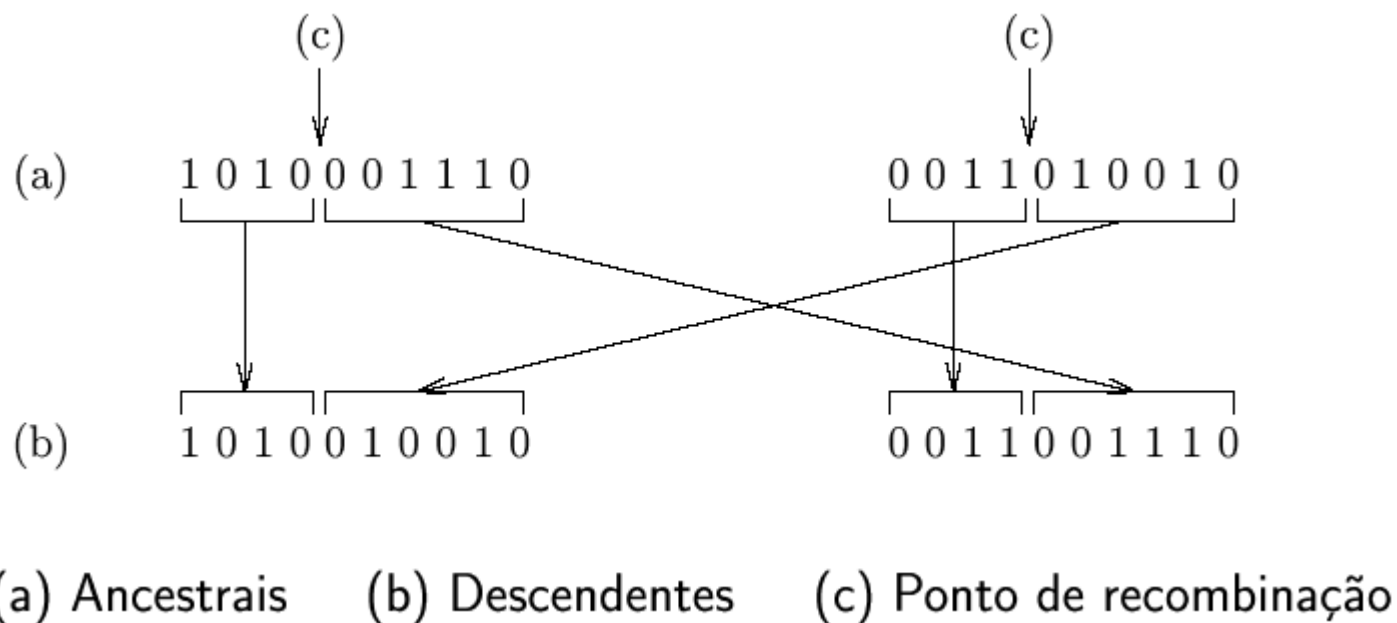
- A seleção serve para escolher **um indivíduo (reprodução assexuada)** ou um **par de indivíduos (reprodução sexuada)** para gerar descendência.
- Para definir os indivíduos da geração seguinte, as soluções com maior valor de retorno da função de aptidão são selecionadas.
- Tendo selecionado os pares, novos indivíduos são gerados por **recombinação e mutação**.

Recombinação (crossover)

- **Recombinação** (ou **cruzamento**) é um operador que simula a troca de material genético entre os ancestrais.
- Recombinação não é sempre aplicada. É feita uma seleção randômica de casais, que sofrem recombinação com uma dada **probabilidade de cruzamento** (entre 0,6 e 1,0).
- Se não for feita recombinação, a reprodução consiste em simples réplica dos pais (2 descendentes).
- Na sua forma mais simples (“single point crossover”), pais tem seus cromossomos cortados em posição aleatória, gerando 2 caudas e 2 cabeças, que são recombinadas.

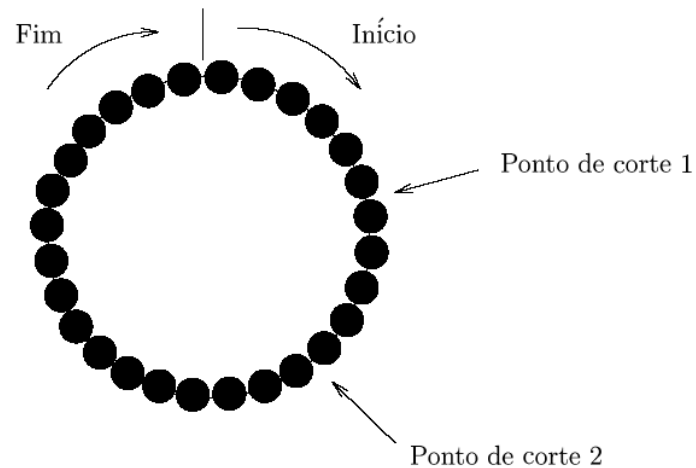
Recombinação (crossover)

Single-point crossover:



Recombinação

- **Multi-point crossover - Recombinação multipontos:** cromossomos vistos como anéis com início e fim unidos. Escolhem-se dois pontos de ruptura do anel e o trecho entre estes pontos é substituído. Padrões são rompidos proporcionalmente ao seu comprimento de definição.



Recombinação

- **Uniform crossover – Recombinação Uniforme:** cada gene dos descendentes é criado copiando o gene correspondente de um dos pais conforme uma máscara gerada aleatoriamente. Onde houver um 1 na máscara, o gene é copiado do ancestral 1, e onde houver um 0, o gene é copiado do ancestral 2. Para gerar o segundo descendente, os ancestrais são trocados de posição.

Máscara de Recombinação	1 0 0 1 0 1 1 1 0 0
Ancestral 1	1 0 1 0 0 0 1 1 1 0
	↓ ↓ ↓ ↓ ↓
Descendente	1 1 0 0 0 0 1 1 1 1
	↑ ↑ ↑ ↑ ↑
Ancestral 2	0 1 0 1 0 1 0 0 1 1

Mutação

- Mutação é aplicada a cada descendente após a recombinação.
- Ela altera randomicamente cada gene com um pequena **probabilidade de mutação** (em geral da ordem de 0,001).
- No caso mais simples, ela implica na simples inversão de um bit (para codificação binária).
- Mutação é tradicionalmente vista como menos importante na evolução do que a recombinação.

Mutação

(a) 1 0 1 0 0 1 0 0 1 1 1 0



(b) 1 0 1 0 1 1 0 0 1 1 1 0

(a) Descendente

(b) Descendente após mutação

Mutação para codificação binária

Mutação

- Se a codificação é por **permutação**, a mutação é feita escolhendo dois números aleatoriamente e trocando-os entre si.
- Se a codificação é por **valores** (por exemplo, números reais), um pequeno valor é adicionado ou subtraído dos genes selecionados.

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 um certo grau de uniformidade.
- Diz-se que um gene convergiu quando 95% da população compartilha o mesmo valor.
- Uma população convergiu quando todos os genes que a compõem convergiram.

Porque AGs funcionam?

- Maioria dos trabalhos em AG procura encontrar regras empíricas para melhorar seu desempenho.
- Não há uma teoria geral que explica de forma completa como e porque a técnica funciona.
- Existem algumas hipóteses bem aceitas:
 - Teorema “Esquema” (schema theorem) de Holland (1975)
 - Hipótese dos Blocos Construtivos de Goldberg (1989)
 - Exploration x Exploitation

Teorema Esquema

- Proposto por Holland em 1975 (“schema theorem”).
- Com o passar das gerações, as soluções boas tendem a compartilhar certas partes de seus cromossomos.
- Estas partes compartilhadas são chamadas de “esquemas” ou “padrões”.
- Padrões com maior aptidão do que a média tendem a se multiplicar exponencialmente ao longo das gerações.
- Padrões com menor aptidão tendem a desaparecer.

Teorema Esquema

- Um esquema é um padrão de valores de genes que podem ser representados em codificação binária por um string contendo os elementos $\{0, 1, \#\}$, onde $\#$ vale 0 ou 1.
- Um cromossomo pode conter um ou mais padrões.
 - Ex.: 1010 contém, entre outros, os padrões 10##, #0#0, ##1# e 101#.
 - A **ordem** do esquema é o número de símbolos diferentes de $\#$ contidos nele (2, 2, 1, 3 no exemplo).
 - O **comprimento de definição** do esquema é o número de símbolos que separam dois elementos mais externos diferentes de $\#$ no cromossomo (2, 3, 1, 3 no exemplo).

Teorema Esquema

- Assume-se que elementos bem sucedidos da população possuam “bons” padrões.
- Holland mostrou que o modo ótimo de explorar o espaço de busca é alocar **oportunidades de reprodução** aos indivíduos de uma população em proporção direta a sua **aptidão relativa** ao resto da população.
- Assim, bons padrões (esquemas) recebem número exponencialmente crescente de chances de reprodução ao longo das gerações.
- Número de esquemas sendo processados em cada geração é da ordem de n^3 , sendo n o tamanho da população.

Hipótese dos Blocos Construtivos

- Proposta por Goldberg em 1989.
- Poder dos AGs reside em ser capaz de encontrar “bons” blocos construtivos.
- **Blocos construtivos** são esquemas (padrões) com **comprimento de definição pequeno**, constituídos de bits que trabalham bem em conjunto e tendem a melhorar a aptidão de um indivíduo.

Hipótese dos Blocos Construtivos

- Um esquema bem sucedido de codificação encoraja a formação de blocos construtivos garantindo que:
 - Genes relacionados estão próximos entre si no cromossomo;
 - Existe pouca interação entre genes.
- **Interação** entre genes significa que a contribuição de um gene para a aptidão do indivíduo depende do valor de outros genes.
 - Ex: morcegos devem emitir gritos de alta frequência e ser capazes de escutar os ecos. Genes para boa audição só aumentam a aptidão do morcego se os genes para produção de gritos agudos estão presentes.
- Segundo esta hipótese, AGs são eficientes quando (e se) satisfazem as duas condições acima.

Exploration x Exploitation

- Ambos os termos são (infelizmente) traduzidos como “exploração” em português.
- **Exploration** é exploração no sentido de investigar áreas novas e ainda desconhecidas no espaço de busca (“explorar o espaço”).
- **Exploitation** é exploração no sentido de aproveitar ao máximo o conhecimento obtido em pontos visitados do espaço de busca (“explorar” no sentido de “tirar proveito”).
- Algoritmos de busca eficientes devem fazer ambas as coisas.

Exploration x Exploitation

- **Busca Aleatória** é boa em exploration, mas não faz exploitation.
- **Método do Gradiente** (subida de montanha) é bom em exploitation mas faz pouca exploration.
- Holland mostrou que AGs fazem ambas as coisas.

Aspectos Práticos de AG

- A implementação prática de um AG requer atenção para várias questões:
 1. Escolha da Função de Aptidão
 2. Problemas de convergência
 3. Escolha da Técnica de Seleção
 4. Lacuna entre gerações (generation gap)

1. Escolha da Função de Aptidão

- Função de Aptidão ideal: suave e regular, de modo que cromossomos com aptidão razoável estejam próximos de cromossomos com aptidão um pouco melhor.
- Recomendável procurar FA que tem poucos pontos de máxima (multimodais) ou que tenham um máximo global bem isolado.
- FA deve refletir o valor do cromossomo de maneira “real”. Por exemplo, se o objetivo é encontrar o ponto de máxima ou mínima de uma função, a função de aptidão será a própria função a otimizar.

1. Escolha da Função de Aptidão

- Em problemas de otimização combinatória (multi-objetivos) o valor “real” de um cromossomo pode não ser uma quantidade útil para guiar a busca.
- FA neste caso deve mostrar a aptidão de um cromossomo para conduzir a busca em direção a um cromossomo válido (se não sabemos onde estão os cromossomos válidos, isto é impossível).
- Muitas vezes FA deve recompensar sub-objetivos.
- Pode-se introduzir também funções de punição (penalty functions), que indicam o quão pobre é um cromossomo. A FA fica sendo uma constante menos a punição.

2. Problemas de Convergência

- **Convergência prematura:** como a população é finita, as vezes indivíduos de alta aptidão (mas não de aptidão ótima) levam AG a convergir para um ponto de máxima local e não global.
- **Convergência lenta:** quando a aptidão dos melhores indivíduos é próxima da média da população (costuma ocorrer perto do fim do algoritmo), a convergência pode ser lenta.
- Para resolver ambos os problemas, é preciso modificar o modo de **seleção** de indivíduos para reprodução.

3. Técnicas de Seleção

- **Seleção** é a tarefa de alocar **oportunidades de reprodução** (“reproductive trials”) a cada indivíduo da população.
- **Oportunidade de reprodução:** valor inteiro derivado da **aptidão** que indica o **número de cópias** que o indivíduo terá no **grupo de reprodutores** (“mating pool”).

3. Técnicas de Seleção

- Método geral de seleção:
 - Indivíduos da população original são copiados para um **grupo de reprodutores** (em geral do tamanho da população original).
 - Indivíduos com **boa aptidão** são selecionados muitas vezes para serem copiados no grupo de reprodutores.
 - Indivíduos com aptidão média são selecionados menos vezes.
 - Indivíduos com baixa aptidão nunca são selecionados.
 - Após a formação do Grupo de Reprodutores, pares de indivíduos deste grupo são escolhidos randomicamente para reprodução, sendo tirados do grupo até que este esteja vazio.

3. Técnicas de Seleção

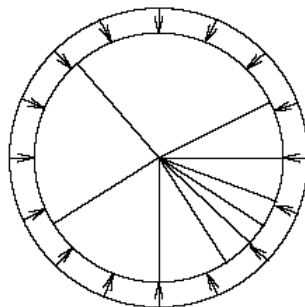
- Métodos de seleção baseados na forma de usar a **aptidão** de cada individuo para calcular valor da **oportunidade de reprodução**:
 - Remapeamento Explícito de Aptidão (Explicit Fitness Remapping)
 - Remapeamento Implícito de Aptidão (Implicit Fitness Remapping)

Remapeamento Explícito de Aptidão

- Método básico de remapeamento: a aptidão de cada indivíduo é remapeada dividindo o **valor da aptidão do indivíduo** pela **aptidão média** da população.
- As **oportunidades de reprodução** são alocadas de modo proporcional a este valor remapeado.
- O resultado da divisão deve ser convertido em inteiro de tal modo a não introduzir uma propensão (“bias”), o que requer um método de amostragem.

Remapeamento Explícito de Aptidão

- Amostragem Universal Estocástica (Stochastic universal sampling) ou Seleção por Roleta:
 - considere um círculo dividido em n regiões (tamanho da população), onde a área de cada região é proporcional à aptidão do indivíduo. Coloca-se sobre este círculo uma "roleta" com n cursores, igualmente espaçados. Após um giro da roleta a posição dos cursores indica os indivíduos selecionados. Os indivíduos cujas regiões possuem maior área terão maior probabilidade de serem selecionados várias vezes.



Remapeamento Explícito de Aptidão

- Seleção por Classificação:
 - Roleta tem problemas quando há grandes diferenças entre os valores de aptidão.
 - A Seleção por Classificação primeiro classifica a população e então atribui a cada cromossomo um valor de aptidão determinado pela sua classificação.
 - O pior terá aptidão igual a 1 , o segundo pior 2 etc. de forma que o melhor terá aptidão igual a N (número de cromossomos na população).
 - Agora todos os cromossomos tem uma chance de serem selecionados. Entretanto, este método pode resultar em menor convergência, porque os melhores cromossomos não se distinguem muito dos outros.

Remapeamento Explícito de Aptidão

- Métodos de remapeamento explícito para evitar convergência prematura:
 - **Escalamento de Aptidão (Fitness Scaling):** define-se um valor máximo de oportunidades de reprodução (típico 2,0) subtraindo uma quantidade fixa da aptidão e dividindo pela média das aptidões modificadas. Aumenta a razão entre a aptidão máxima e a média.
 - **Janelamento de Aptidão (Fitness Windowing):** subtrai o valor da mínima aptidão observada nas últimas n gerações e divide pela média das aptidões modificadas.
 - **Ordenamento de Aptidão (Fitness Ranking):** ordenamento linear ou exponencial dos indivíduos de acordo com seu valor bruto de aptidão. Normaliza a razão entre a aptidão máxima e a média e é menos sensível a indivíduos com valores extremos de aptidão (seleção por classificação é um exemplo).

Remapeamento Implícito de Aptidão

- Grupo de reprodução é preenchido sem o passo intermediário de remapear a aptidão.
 - **Seleção por Torneio (Tournament Selection):** seleciona aleatoriamente um grupo de n indivíduos da população original e copia o indivíduo com melhor aptidão no grupo de reprodução. Todos os indivíduos são devolvidos à população original. O processo se repete até encher o grupo de reprodução.
 - **Seleção por Torneio Probabilista (Probabilistic Tournament Selection):** idem ao anterior, mas o melhor indivíduo vence o torneio com probabilidade p ($0,5 < p < 1,0$).

4. Lacuna entre Gerações

- Lacuna entre gerações (generation gap): proporção dos indivíduos da população que são substituídos em cada geração (taxa substituídos / total).
- Trabalhos mais antigos usam um valor de 1 para o gap (toda a população é substituída por descendentes em uma geração).
- **Substituição de estado estacionário** (steady-state replacement): somente alguns indivíduos são substituídos em cada geração.
- **Elitismo**: pelo menos uma cópia sem alterações da melhor solução da geração anterior é passada para a nova população, de forma que a melhor solução possa sobreviver às sucessivas gerações.
- Este pode ser um modelo melhor do que ocorre na natureza (pelo menos em espécies superiores).
- Pais podem auxiliar os filhos, mas também competem com eles.

4. Lacuna entre Gerações

- **Substituição de estado estacionário** requer seleção não só de reprodutores, mas de indivíduos a morrer.
- Métodos mais usuais:
 - Seleção de reprodutores por aptidão e seleção de indivíduos a morrer de forma aleatória;
 - Seleção de reprodutores de forma aleatória e seleção de indivíduos a morrer por aptidão inversa (mais baixa);
 - Seleção de reprodutores por aptidão e seleção de indivíduos a morrer por aptidão inversa.

AG e Computação Evolutiva

- AG é uma das técnicas da **Computação Evolutiva**, que inclui ainda:
 - **Estratégia Evolutiva:** dá ênfase na auto-adaptação (recombinação existe mas é secundária);
 - **Programação Genética:** indivíduos são programas armazenados na forma de árvores sintáticas. A recombinação se dá por troca de sub-árvores entre indivíduos e não há mutação;
 - **Programação Evolutiva:** previsão do comportamento de máquinas de estado finitas. Só usa seleção e mutação (sem recombinação).

Aplicações

- Otimização de funções numéricas descontínuas, multimodais e ruidosas;
- Processamento de imagens;
- Otimização combinatória: caixeiro viajante, escalonamento de tarefas em fábricas;
- Combinação de otimização de funções com otimização combinatória: problemas de Engenharia (como projeto de uma ponte com taxa otimizada entre força e peso), treinamento de redes neurais;
- Determinação do corte ideal de tecido para fabricar roupas (com perda mínima de material);
- Aprendizado de máquina (machine learning): classificadores;
- Controle de processos: geração de regras de controle;
- Balanceamento de carga em sistemas computacionais multiprocessados;
- Etc;

Bibliografia básica

- J.H. Holland. Adaptation in Natural and Artificial Systems. MIT Press, 1975.
- D. Beasley. An Overview of Genetic Algorithms: Part 1, Fundamentals
- D. Beasley. An Overview of Genetic Algorithms: Part 2, Research Topics
- D. Whitley. A Genetic Algorithm Tutorial.
- <http://www.cidase.com/webizu/professor/ga/>