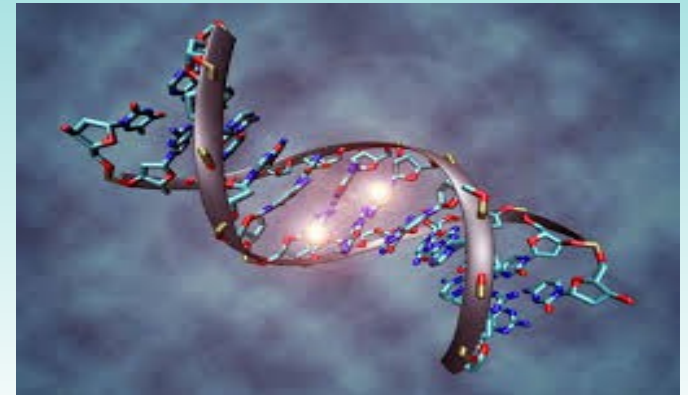




Algoritmos Genéticos

(uma breve introdução)

Prof^a Carine Webber



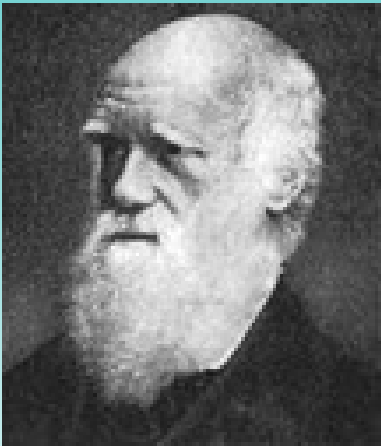
Algoritmos Genéticos



- Algoritmos que implementam uma técnica de busca em um espaço de soluções.
- Resolvem problemas de otimização.
- Criados a partir da teoria da evolução das espécies iniciada por Charles Darwin.
- Foram desenvolvidos por John Holland (1975) e sua equipe.

Relembrando... a Teoria da Evolução

- 1859 - Charles Darwin publica o livro “*A Origem das Espécies*”:



Charles
Darwin

“As espécies evoluem pelo princípio da seleção natural e sobrevivência do mais apto.”

Princípios da Seleção Natural



A seleção natural é um processo que ocorre ao longo de várias gerações.

Os indivíduos de uma mesma espécie apresentam variações em todos os caracteres, não sendo, portanto, idênticos entre si.

Todo organismo tem grande capacidade de reprodução, produzindo muitos descendentes.

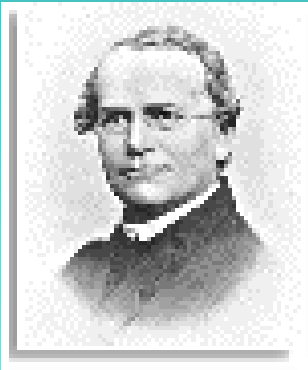
O número de indivíduos de uma espécie é mantido mais ou menos constante ao longo das gerações.

Na luta pela vida, organismos com variações favoráveis às condições do ambiente onde vivem têm maiores chances de sobreviver, quando comparados aos organismos com variações menos favoráveis.

Os organismos com essas variações vantajosas têm maiores chances de deixar descendentes.

Assim, ao longo das gerações, a atuação da seleção natural sobre os indivíduos mantém ou melhora o grau de adaptação destes ao meio.

Teoria da Evolução



Gregor
Mendel

- 1865 - Gregor Mendel apresenta experimentos do cruzamento genético de ervilhas.
 - ◆ Pai da genética.
- A Teoria da Evolução se desenvolveu pela integração da seleção natural com a Genética.



Algoritmos Genéticos



Um algoritmo genético pode ser definido como um procedimento iterativo de busca (otimização) inspirado nos mecanismos evolutivos biológicos.

John Holland (1975)

A computação evolutiva é o nome utilizado para descrever a linha de pesquisa que trata de algoritmos evolutivos.

Otimização



- É a busca da melhor solução para um dado problema.
 - ◆ Consiste em tentar várias soluções e usar a informação obtida para conseguir soluções cada vez melhores.
- Exemplo de otimização:
 - ◆ Sintonizar rádio
 - ◆ Configurar rede
 - ◆ Configurar hardware

Otimização



- As técnicas de otimização, geralmente, apresentam:
 - Espaço de busca: onde estão todas as possíveis soluções do problema.
 - ♦ Função objetivo: utilizada para avaliar as soluções produzidas, associando a cada uma delas uma nota.

Características dos Algoritmos Genéticos



- Não é uma abordagem determinística.
- Trabalha com uma população de soluções simultaneamente.
- Utiliza apenas informações de custo e recompensa. Não requer nenhuma outra informação auxiliar.

Características dos Algoritmos Genéticos



- São fáceis de serem implementados em computadores.
- Adaptam-se bem a computadores paralelos.
- São facilmente hibridizados com outras técnicas.
- Funcionam com parâmetros contínuos ou discretos.

Algoritmos Genéticos



- AG manipula uma população de indivíduos.
- Indivíduos são possíveis soluções do problema.
- Os indivíduos são combinados (crossover) uns com os outros, produzindo filhos que podem sofrer ou não mutação.
- As populações evoluem através de sucessivas gerações até encontrar a solução ótima.

Aplicações



- Em problemas difíceis de otimização, quando não existe nenhuma outra técnica específica para resolver o problema.
- Otimização de funções numéricas em geral
- Otimização combinatória
 - ♦ Problema do caixeiro viajante
 - ♦ Problema de empacotamento
 - ♦ Alocação de recursos (*job shop schedulling*)
- Aprendizado de Máquina

Terminologia



- **Indivíduo**
 - Simples membro da população.
- **Cromossomo :**
 - Coleção de genes
 - Estrutura de dados que codifica a solução de uma problema.
- **Genótipo**
 - Na biologia, representa a composição genética contida no Genoma.
 - Representa a informação contida no cromossomo.

Terminologia



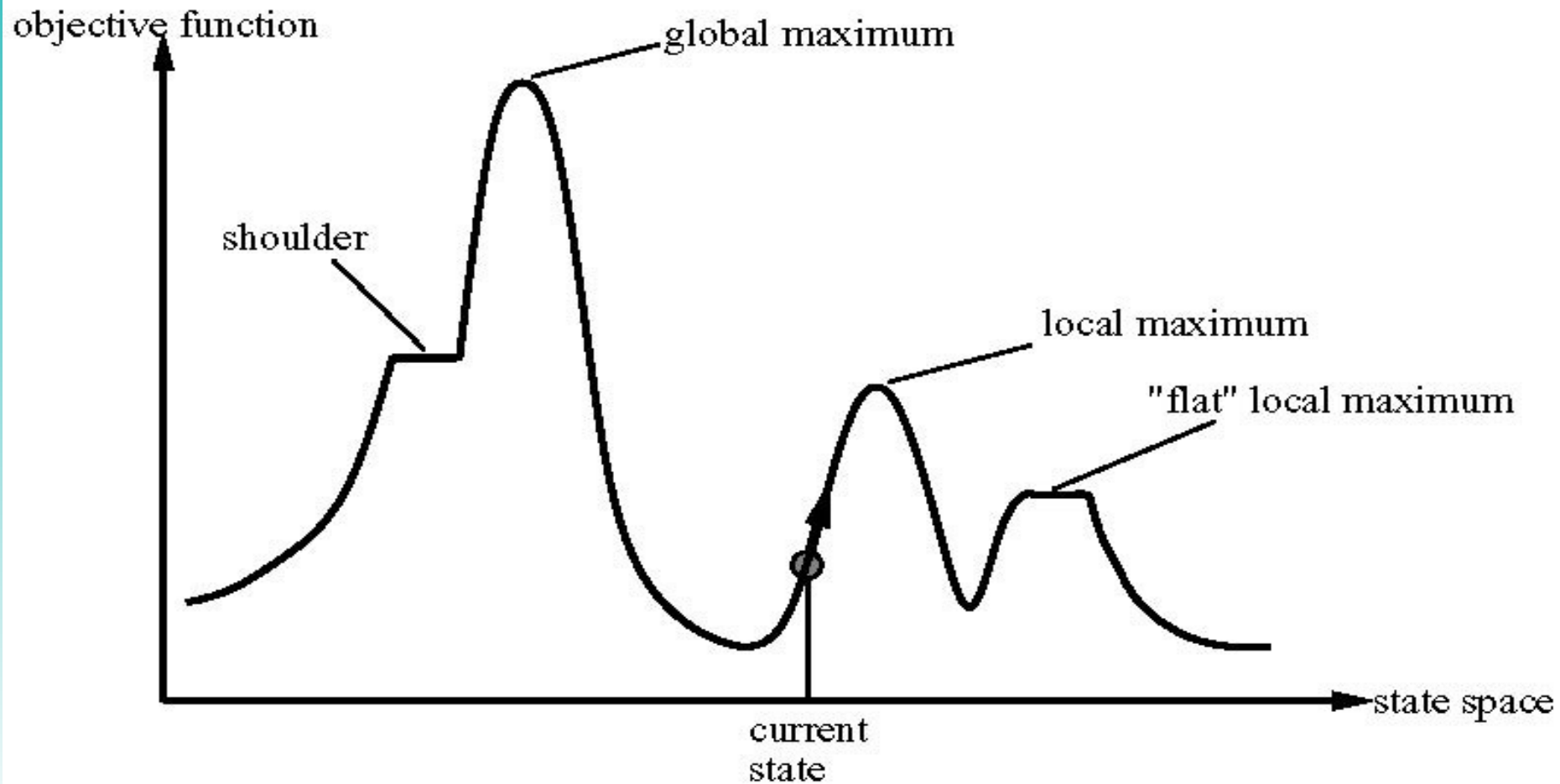
- **Fenótipo:**
 - Objeto ou estrutura construída a partir das informações do genótipo.
 - É o cromossomo decodificado.
 - Exemplo: Se o cromossomo codifica as dimensões de um edifício, então o fenótipo é a aparência do edifício construído.
- **Gene:**
 - Codifica um simples parâmetro do problema

Terminologia



- **Alelos:**
 - Valores que o gene pode assumir.
 - Ex.: Um gene representando a cor de um objeto pode ter alelos como azul, preto, verde etc...
- **Seleção Natural:**
 - Através de uma função de avaliação, objetivo ou *fitness* calcula-se a adaptabilidade, qualidade ou *fitness* de cada indivíduo.
 - Indivíduos com maiores valores de *fitness* tem maior probabilidade de sobrevivência e de reprodução (herança genética).

Função objetivo



indivíduos

Algoritmo Genético Clássico



1. Gerar a população inicial.
2. Avaliar cada indivíduo da população.
3. Enquanto critério de parada não for satisfeito faça
 - 3.1 Selecionar os indivíduos mais aptos.
 - 3.2 Criar novos indivíduos aplicando os operadores crossover e mutação.
 - 3.3 Armazenar os novos indivíduos em uma nova população.
 - 3.4 Avaliar cada cromossomo da nova população.

Problema 1

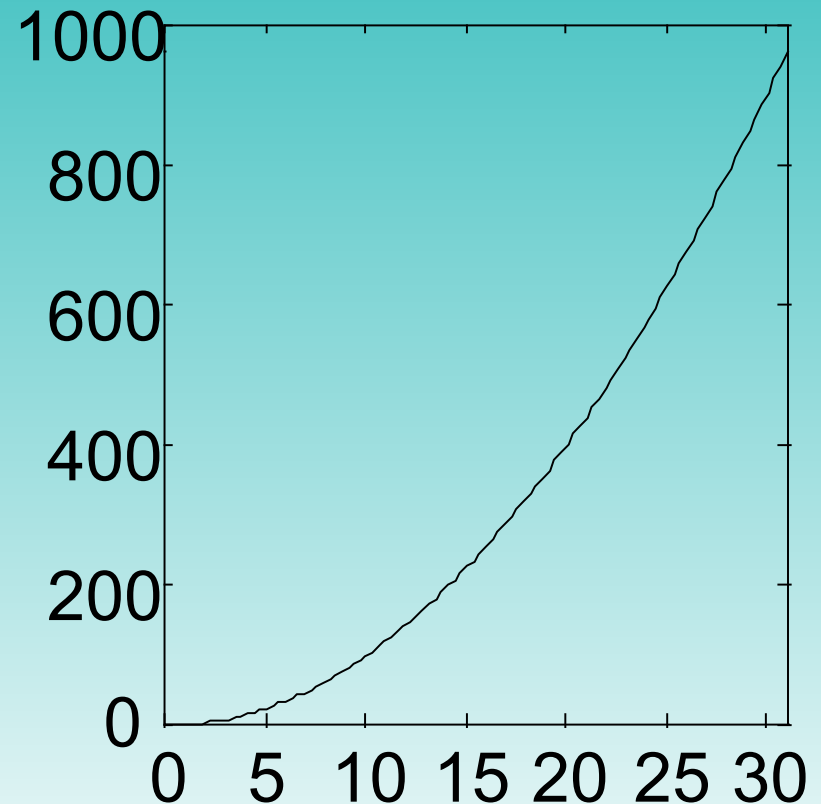
Problema: Use um AG
para encontrar o ponto
máximo da função:

$$f(x) = x^2$$

com x sujeito as seguintes
restrições:

$$0 \leq x \leq 31$$

x é inteiro



Indivíduo



- **Cromossomo**
 - Estrutura de dados que representa uma possível solução para o problema.
 - Os parâmetros do problema de otimização são representados por cadeias de valores.
 - Exemplos:
 - Vetores de reais, (2.345, 4.3454, 5.1, 3.4)
 - Cadeias de bits, (111011011)
 - Vetores de inteiros, (1,4,2,5,2,8)
 - ou outra estrutura de dados.

Indivíduo



- **Aptidão**
 - Nota associada ao indivíduo que avalia quão boa é a solução por ele representada.
- **Aptidão pode ser:**
 - Igual a função objetivo (raramente usado na prática).
 - Resultado do **escalonamento** da função objetivo.
 - Baseado no **ranking** do indivíduo da população.

Cromossomo do Problema 1



- Cromossomos binários com 5 bits:
 - 0 = 00000
 - 31 = 11111
- Aptidão (*Fitness*)
 - Neste problema, a aptidão pode ser a própria função objetivo.
 - Exemplo:

$$\text{aptidão}(00011) = f(3) = 9$$

Seleção



- **Seleção**
 - Imitação da seleção natural.
 - Os melhores indivíduos (maior aptidão) são selecionados para gerar filhos através de crossover e mutação.
 - Dirige o AG para as melhores regiões do espaço de busca.
- **Tipos mais comuns de seleção**
 - Proporcional a aptidão.
 - Torneio.

População Inicial do Problema 1



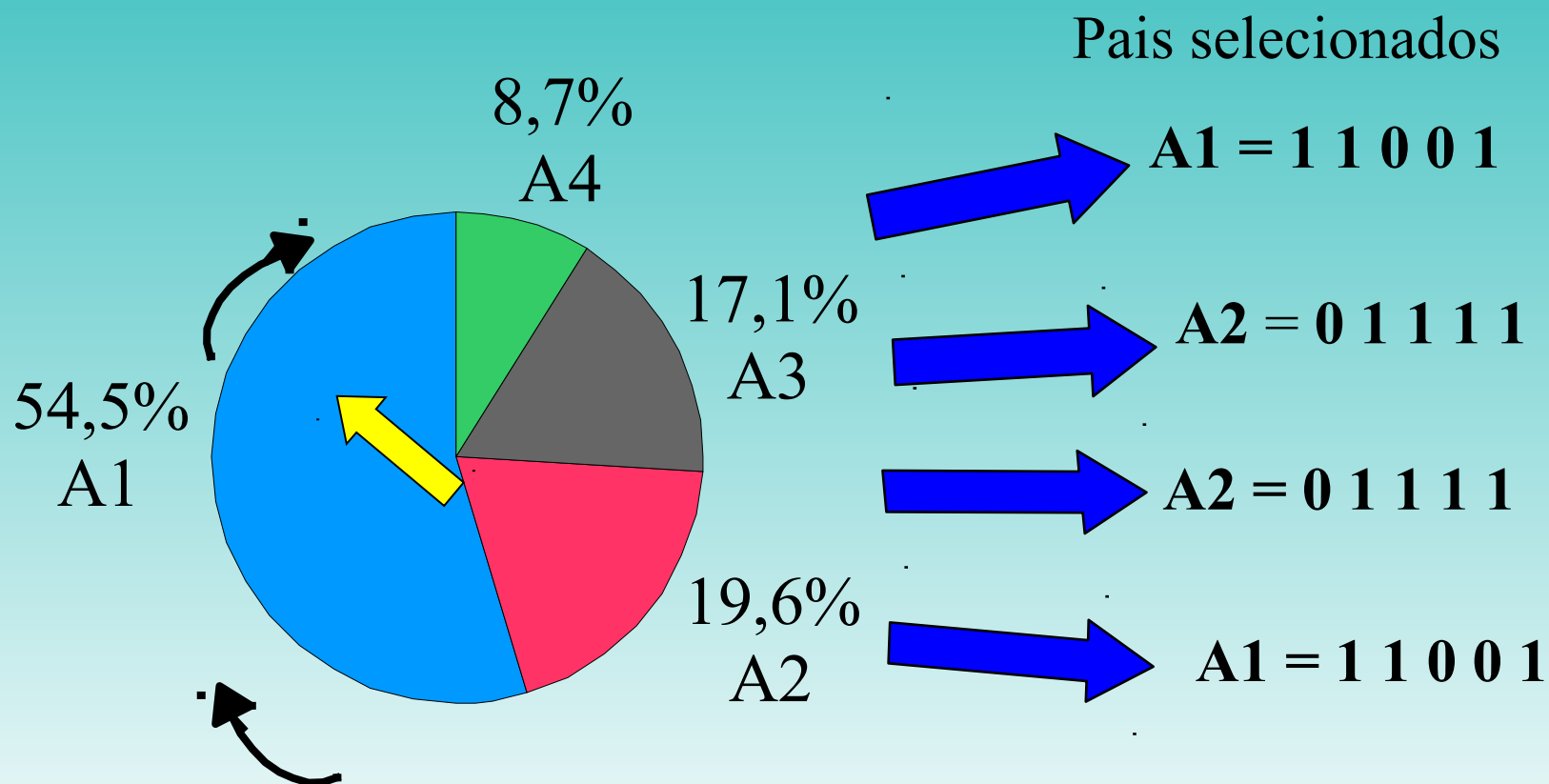
É aleatória (mas quando possível, o conhecimento da aplicação pode ser utilizado para definir população inicial)

Pop. inicial	cromossomos	x	$f(x)$	Prob. de seleção
	$A_1 = 1\ 1\ 0\ 0\ 1$	25	625	54,5%
	$A_2 = 0\ 1\ 1\ 1\ 1$	15	225	19,6%
	$A_3 = 0\ 1\ 1\ 1\ 0$	14	196	17,1%
	$A_4 = 0\ 1\ 0\ 1\ 0$	10	100	8,7%

Probabilidade de seleção
proporcional a aptidão

$$p_i = \frac{f(x_i)}{\sum f(x_k)}$$

Seleção proporcional a aptidão (Roleta)



Seleção por Torneio



- Escolhe-se n (tipicamente 2) indivíduos aleatoriamente da população e o melhor é selecionado.

Crossover e Mutação

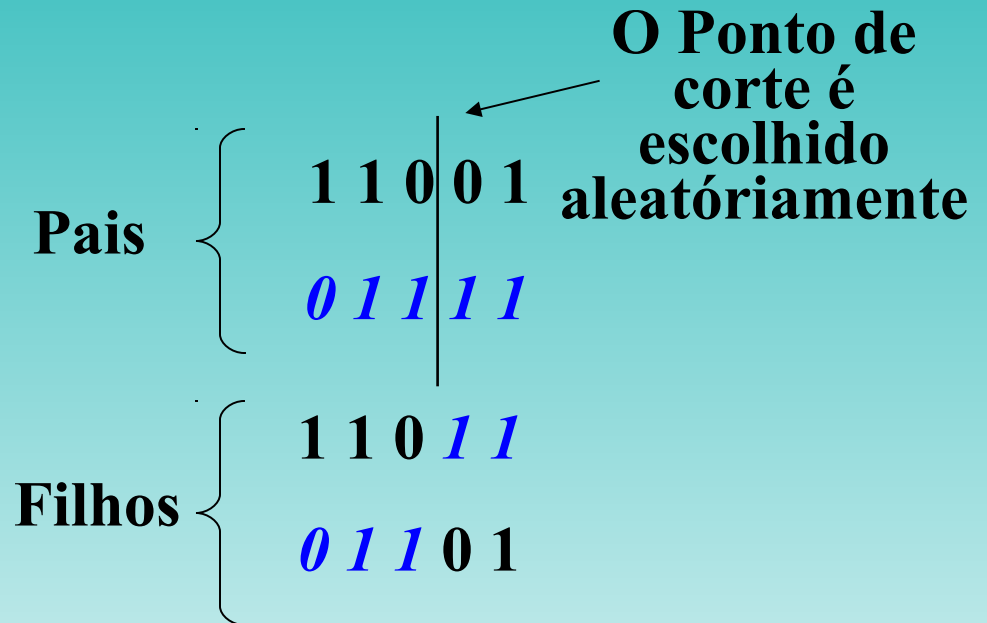


- Combinam pais selecionados para produção de filhos.
- Principais mecanismos de busca do AG.
- Permite explorar áreas desconhecidas do espaço de busca.

Crossover de 1 ponto



O crossover é aplicado com uma dada probabilidade denominada *taxa de crossover* (60% a 90%)



Se o crossover é aplicado, os pais trocam seus bits gerando dois filhos, caso contrário os dois filhos serão cópias exatas dos pais.

Mutação



Mutação inverte os valores dos bits.

A mutação é aplicada com dada probabilidade, denominada *taxa de mutação* (~1%), em cada um dos bits do cromossomo.

**Antes da
mutação** 0 1 1 0 1

Depois 0 0 1 0 1

Aqui, apenas o 2o.bit
passou no teste de
probabilidade

A taxa de mutação não deve ser nem alta nem baixa, mas o suficiente para assegurar a diversidade de cromossomos na população.

A primeira geração do Problema 1



Pais

Filhos

A1 = 1 1 0 | 0 1
A2 = 0 1 1 | 1 1

crossover



1 1 0 1 1
0 1 1 0 1

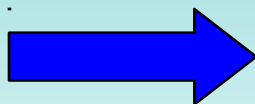
mutação



1 1 0 1 1
0 0 1 0 1

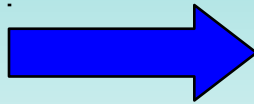
A2 = 0 1 1 1 | 1
A1 = 1 1 0 0 | 1

crossover



0 1 1 1 1
1 1 0 0 1

mutação



1 0 1 1 1
1 1 0 0 1

Nova
pop.

A primeira geração do Problema 1 (II)



cromossomos	x	$f(x)$	prob. de seleção
1 1 1 0 1 1	27	729	29,1%
2 1 1 0 0 1	25	625	24,9%
3 1 1 0 0 1	25	625	24,9%
4 1 0 1 1 1	23	529	21,1%

As demais gerações do Problema 1



Segunda
Geração

		x	$f(x)$
1	1 1 0 1 1	27	729
2	1 1 0 0 0	24	576
3	1 0 1 1 1	23	529
4	1 0 1 0 1	21	441

Terceira
Geração

		x	$f(x)$
1	1 1 0 1 1	27	729
2	1 0 1 1 1	23	529
3	0 1 1 1 1	15	225
4	0 0 1 1 1	7	49

As demais gerações do Problema 1 (II)



Quarta
Geração

		x	$f(x)$
1	1 1 1 1 1	31	961
2	1 1 0 1 1	27	729
3	1 0 1 1 1	23	529
4	1 0 1 1 1	23	529

Quinta
Geração

		x	$f(x)$
1	1 1 1 1 1	31	961
2	1 1 1 1 1	31	961
3	1 1 1 1 1	31	961
4	1 0 1 1 1	23	529

Outros Crossover's



- Crossover de 2-pontos

pai_1	010	011000	101011
pai_2	001	001110	001101
$filho_1$	010	001110	101011
$filho_2$	001	011000	001101

Considerado melhor que o crossover de 1 ponto.

Crossover de n-Pontos



pai_1	101	010010	01010	01001
pai_2	001	001110	00110	11100
$filho_1$	1010011100	10101011	001	
$fillho_2$	0010100100	00110011	00	

Crossover de 4-pontos

Crossover Uniforme



Máscara de
bits aleatória

1 1 0 1 0 1 1 0 1 0

*pai*₁ 1 1 1 0 1 1 0 1 1 0



*filho*₁ 1 1 1 0 0 1 0 1 1 0



*pai*₂ 0 1 1 0 0 0 1 1 0 0

O *filho*₁ tem 50% de chance de levar um bit do *pai*₁ e 50% de chance de levar um bit de *pai*₂

O *filho*₂ leva o que sobra de *pai*₁ e *pai*₂

Elitismo



- O crossover ou mutação podem destruir o melhor indivíduo.
- Por que perder a melhor solução encontrada?
- Elitismo transfere a cópia do melhor indivíduo para a geração seguinte.

Critérios de Parada



- Número de gerações.
- Encontrou a solução (quando esta é conhecida).
- Perda de diversidade.
- Convergência
 - nas últimas k gerações não houve melhora da aptidão dos indivíduos.

Problema do Caixeiro Viajante



Suponha que um caixeiro deva partir de sua cidade, visitar clientes em outras 99 cidades diferentes, e então retornar a sua cidade.

Dadas as coordenadas das 100 cidades, descubra o percurso de menor distância que passe uma única vez por todas as cidades e retorne a cidade de origem.

Codificação



Representação interna: cada cromossomo conterá todos os números de 1 a 100, sendo um número associado a cada cidade e a ordem de aparecimento dos números no cromossomo indica o percurso, sendo necessário que a última e a primeira cidade sejam as mesmas.

Codificação



Número de percursos possíveis: 99!

Função objetivo (fitness): inverso da distância associada a cada percurso.

Solução ótima: desconhecida devido a impossibilidade de testar todas as combinações.

Mutação: sorteio de duas cidades para troca de posição

Taxa de mutação: 1%

Crossover: troca de partes do cromossomo mas com a substituição das cidades repetidas pelas ausentes.

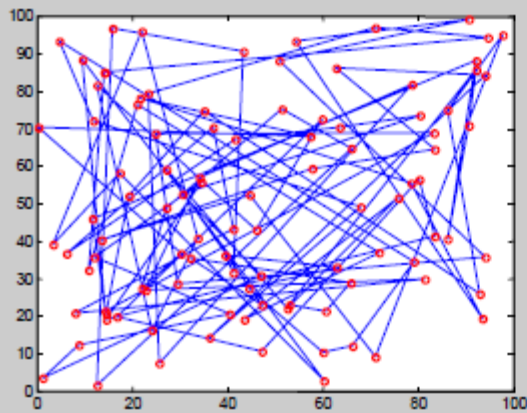
Taxa de crossover: 60%

Tipo de seleção: 50% dos melhores.

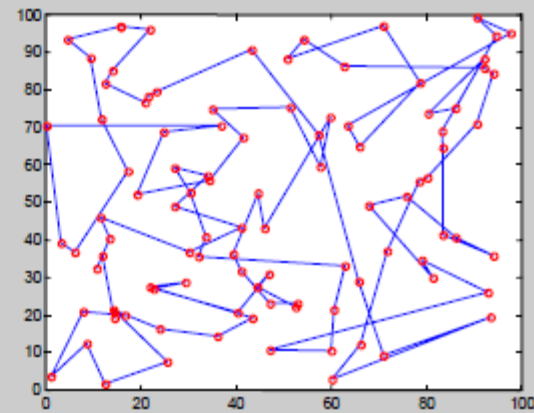
Resultados

Número de indivíduos testados: 400.000

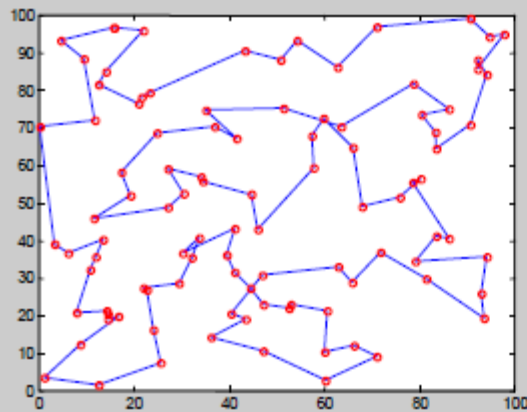
Melhor indivíduo na população inicial



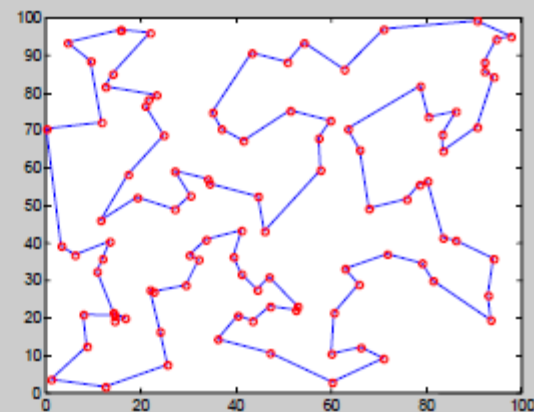
Melhor indivíduo após 500 gerações



Melhor indivíduo após 2000 gerações



Melhor indivíduo após 4000 gerações



Aplicações



- **Planejamento:**
 - Caixeiro viajante
 - Roteamento de veículos
 - Problema de transporte: distribuição de mercadorias para diversos clientes.
 - Robótica: um caminho seguro e sem colisões para um robô.
 - Sequenciamento de tarefas

Aplicações



- **Planejamento**

- Job shop scheduling : alocação de tarefas em máquinas em uma sequencia que permite que todas as tarefas sejam executadas com custo mínimo.
- Problema da Agenda
- Empacotamento (problema da mochila)
- Processamento computacional: alocação de processos em diferentes estações ou processadores.

Aplicações



- **Projetos:**

- Sistemas inteligentes: definição de uma arquitetura e/ou parâmetros de redes neurais, sistemas nebulosos, autômatos celulares e sistemas imunológicos, dentre outros.
- Aplicações em Engenharia: redes de telecomunicações, projetos estruturais, projeto de aeronaves, teste e diagnóstico de falhas, etc.
- Processamento de sinais: otimização do projeto de sistemas de processamento de sinais e desenho de circuitos integrados.

Analise a seguinte questão



Quando devemos utilizar

Hill Climbing (Subida de Encosta)

Simulated Annealing (Resfriamento Simulado) e

Algoritmos Genéticos?

Relembrando



Hill-Climbing: Subida da Encosta ou Gradiente Ascendente

- só faz modificações que melhoram o estado atual.

Simulated Annealing: Resfriamento Simulado

- pode fazer modificações que pioram o estado temporariamente, para possivelmente melhorá-lo no futuro.

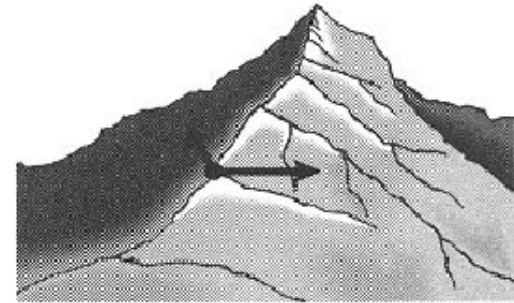
Questões



Qual dos algoritmos
possibilita a saída
deste máximo local?



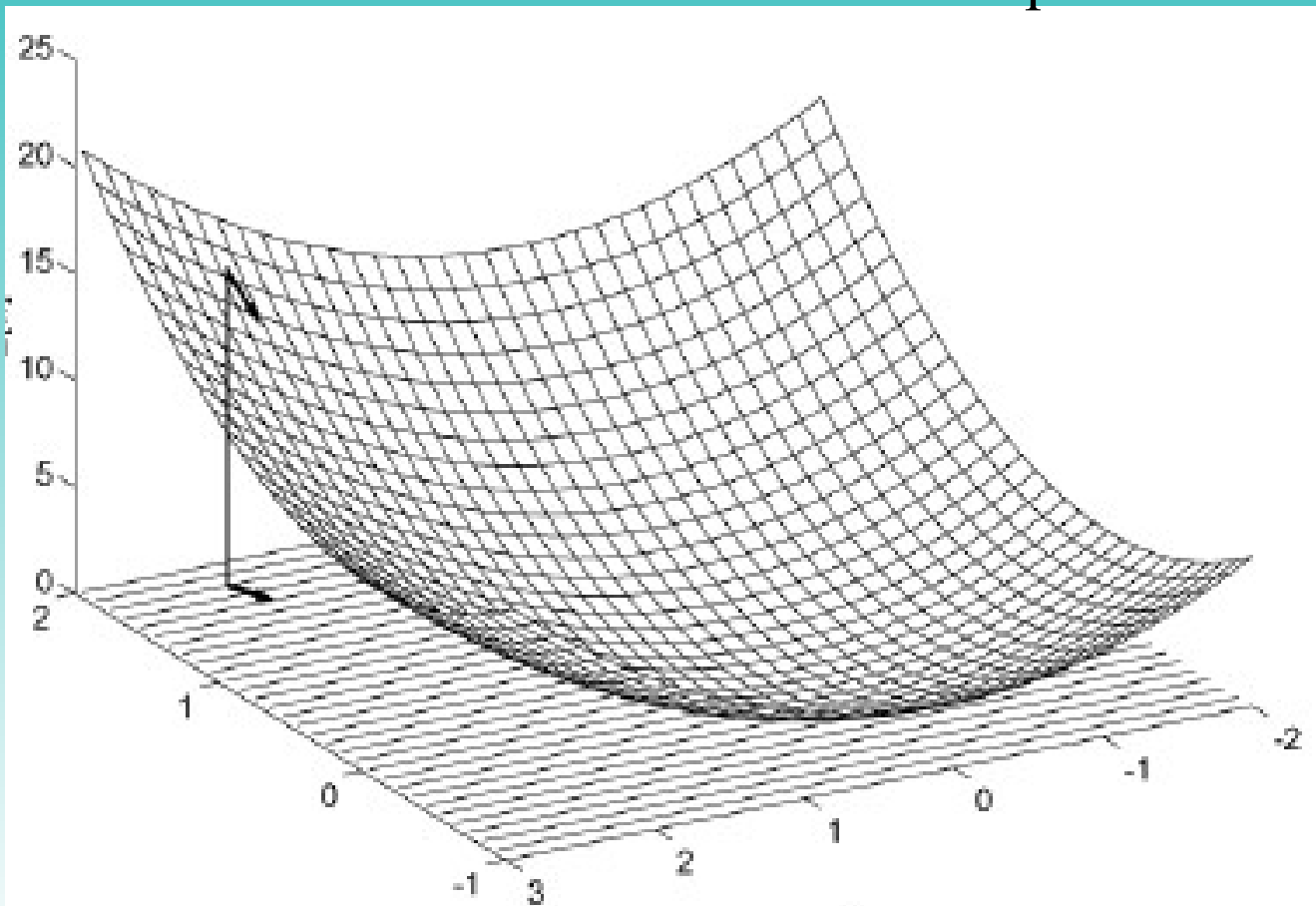
Qual dos algoritmos
possibilita a saída
deste platô?



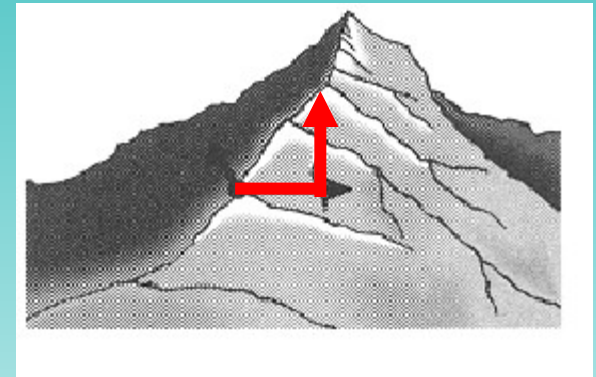
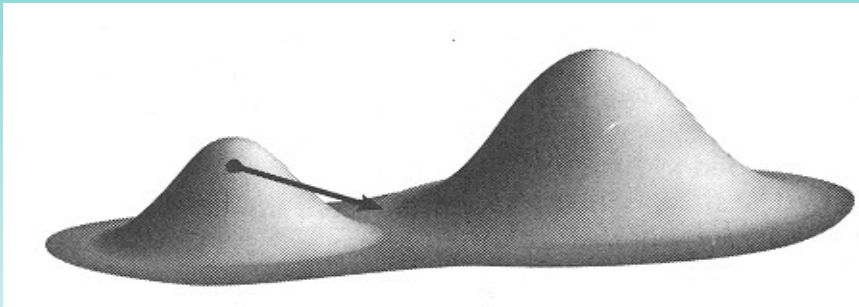
Qual dos algoritmos
possibilita a saída
desta subida
íngreme?

O que poderia ser usado neste exemplo?

Busca por mínimos



Análise dos exemplos



Exercícios



- A primeira lista está disponível na pasta ACERVO.
- As respostas devem ser enviadas via webfólio.