

# GT-JeDi - Curso de Desenv. de Jogos

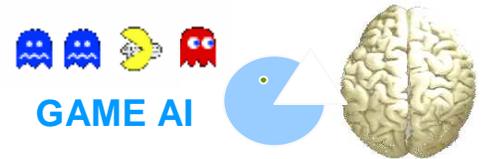
## *IA para Jogos*



Gustavo Pessin

2007

# Algoritmos Genéticos

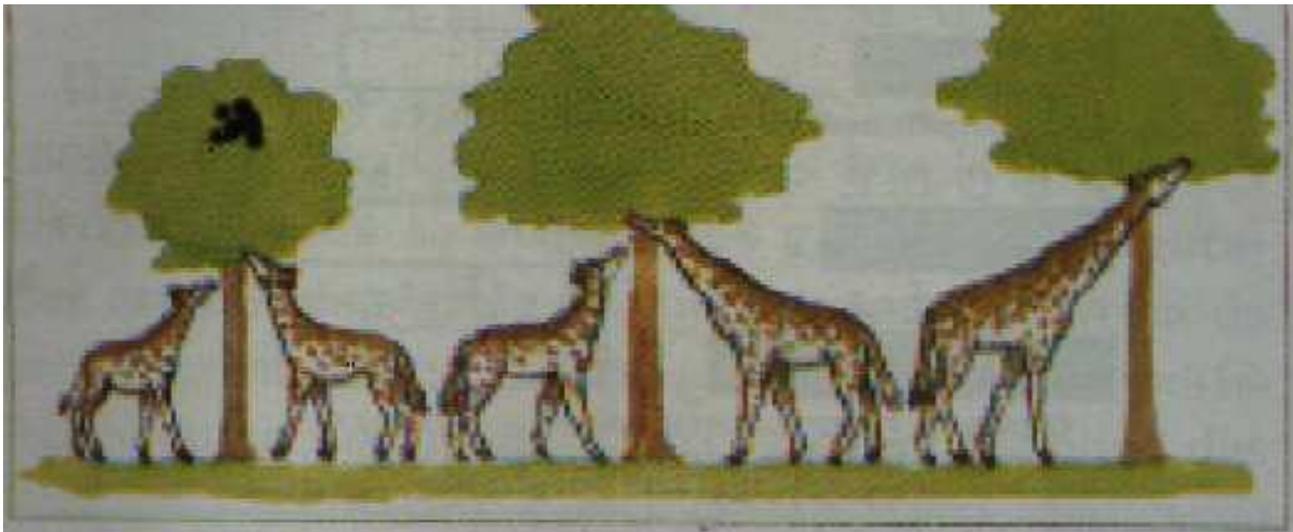


- Cronograma
  - Base conceitual
  - Exemplo: Achando o máximo de uma função...
  - Como criar uma pequena aplicação:
    - Exercício-Exemplo [Animal selvagem...]
  - Exemplo em jogo: Lunar Lander
  - Exemplo: Caixeiro Viajante [pcv]
  - Caminhar robótico (Milton)
  - Cercando objetivos (Robombeiros)

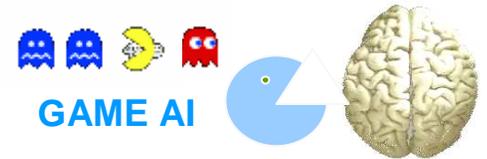
# Algoritmos Genéticos



- Técnicas de busca e otimização.
- Metáfora da teoria da evolução das espécies iniciada pelo Fisiologista e Naturalista inglês Charles Darwin.
- Desenvolvido por John Holland (1975).
- Popularizado por David Goldberg (1989).



# Algoritmos Genéticos



## ➤ Teoria da Evolução

➤ 1859 - Charles Darwin publica o livro

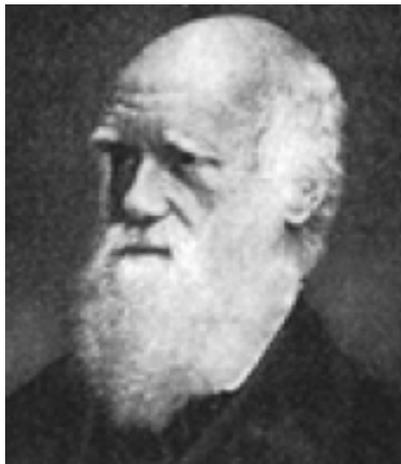
### ➤ “A Origem das Espécies”

➤ *“As espécies evoluem pelo principio da seleção natural e sobrevivência do mais apto.”*

➤ *“Quanto melhor um indivíduo se adaptar ao seu meio ambiente, maior será sua chance de sobreviver e gerar descendentes”*

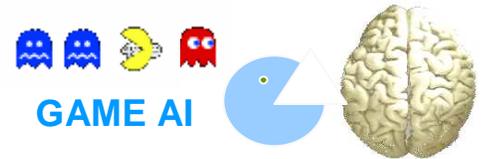
➤ Gregor Mendel, em 1865, apresenta experimentos do cruzamento genético de ervilhas, “Pai da genética”.

➤ Teoria da evolução => conceituação integrada da seleção natural com a genética.



Charles Darwin

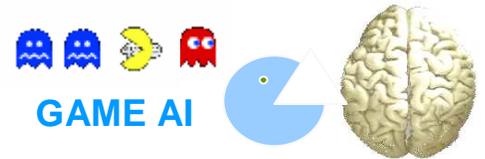
# Algoritmos Genéticos



## ➤ 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
  - *Telespectador através de ajuste na antena da televisão otimiza a imagem buscando várias soluções até alcançar uma boa imagem.*

# Algoritmos Genéticos



## ➤ Otimização

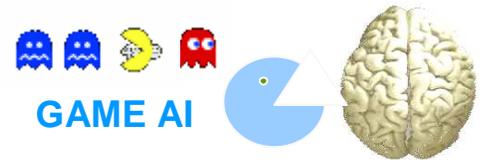
➤ As técnicas de otimização geralmente apresentam:

➤ Espaço de busca, onde estão todas as possíveis soluções do problema;

➤ Função objetivo (**fitness**): utilizada para avaliar as soluções produzidas, associando a cada solução uma nota.

➤ Em termos matemáticos:

➤ Otimização = achar a solução que corresponda ao ponto de máximo ou mínimo de uma função.



## ➤ Algoritmo genético típico

---

Seja  $S(t)$  a população de cromossomos na geração  $t$ .

$t \leftarrow 0$

inicializar  $S(t)$

avaliar  $S(t)$

**enquanto** o critério de parada não for satisfeito **faça**

$t \leftarrow t + 1$

    selecionar  $S(t)$  a partir de  $S(t-1)$

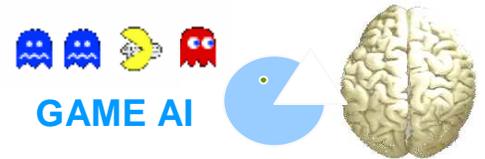
    aplicar *crossover* sobre  $S(t)$

    aplicar mutação sobre  $S(t)$

    avaliar  $S(t)$

**fim enquanto**

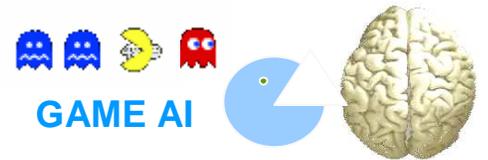
# Algoritmos Genéticos



## ➤ Características

- São algoritmos estocásticos (não é determinístico).
  - **Estocásticos são padrões que surgem através de eventos aleatórios.**
- Trabalha com uma população de soluções simultaneamente.
- Utiliza apenas informações de custo e recompensa, não requer nenhuma outra informação auxiliar (como por exemplo o gradiente).
- São fáceis de serem implementados em computadores.
- São facilmente hibridizados com outras técnicas.
- Funcionam com parâmetros contínuos ou discretos.

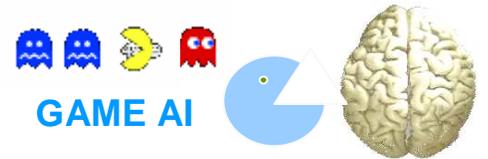
# Algoritmos Genéticos



## ➤ Conceitos

- 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.

# Algoritmos Genéticos



## ➤ Dificuldades

- Como representar os os parâmetros que definem o problema?
  - Usualmente trabalha com strings de bits
- Quem é a população inicial?
- Como definir a função objetivo (fitness)?

# Algoritmos Genéticos

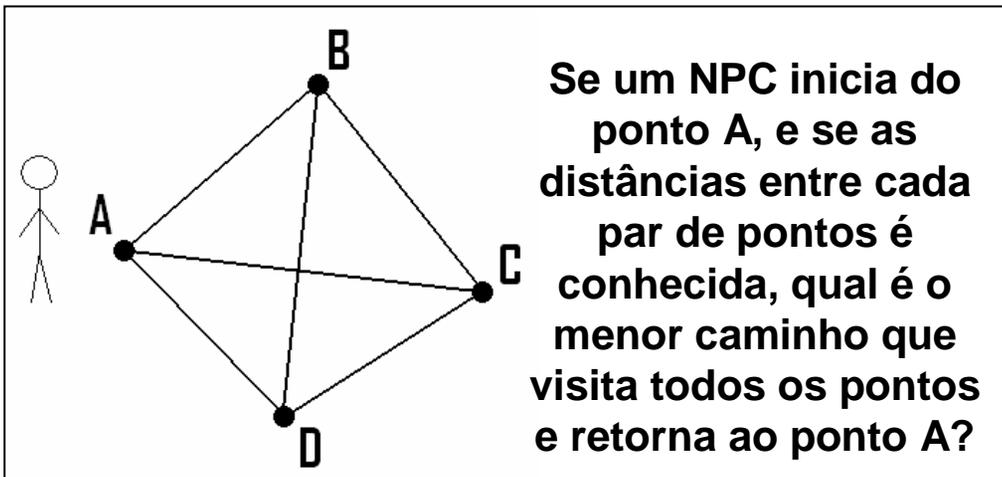


## ➤ Aplicações

➤ Em problemas difíceis de otimização

➤ Quando não existe técnica específica para resolver o problema.

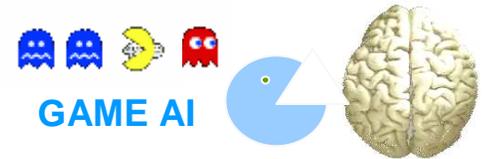
➤ Como o problema NP-completo do caixeiro viajante



Se um NPC inicia do ponto A, e se as distâncias entre cada par de pontos é conhecida, qual é o menor caminho que visita todos os pontos e retorna ao ponto A?

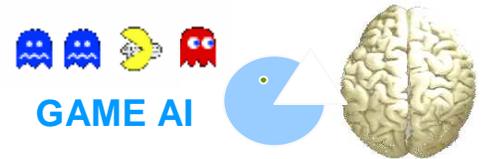
A solução mais direta (simples de implementar) consiste em tentar todas as combinações possíveis, de modo a verificar no final qual é o caminho de menor custo. Dado que a quantidade de combinações é fatorial de número de nós, tal solução é impraticável.

# Algoritmos Genéticos



Nós	Fatorial
1	1
2	2
3	6
4	24
5	120
6	720
7	5.040
8	40.320
9	362.880
10	3.628.800
11	39.916.800
12	479.001.600
13	6.227.020.800
14	87.178.291.200
15	1.307.674.368.000
16	20.922.789.888.000
17	355.687.428.096.000
18	6.402.373.705.728.000
19	121.645.100.408.832.000
20	2.432.902.008.176.640.000

# Algoritmos Genéticos



## ➤ Terminologia

➤ **Indivíduo** (simples membro da população).

### ➤ **Cromossomo e Genoma**

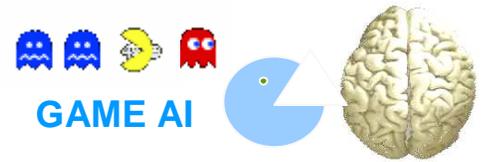
➤ Coleção de genes

➤ Estrutura de dados que codifica a solução de um problema.

### ➤ **Genótipo**

➤ Na biologia, representa a composição genética do organismo, em AGs, representa a informação contida no cromossomo.

# Algoritmos Genéticos



## ➤ Terminologia

### ➤ Gene

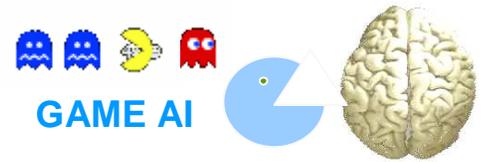
➤ Codifica um simples parâmetro do problema

### ➤ Alelos

➤ Valores que o gene pode assumir.

➤ Um gene representando a cor de um objeto pode ter alelos como azul, preto, verde etc...

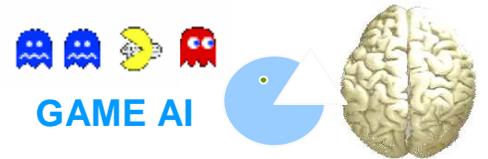
# Algoritmos Genéticos



## ➤ **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)
- Aptidão de um indivíduo => Resultado do fitness

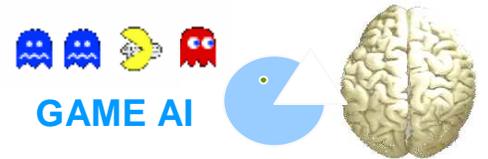
# Algoritmos Genéticos



## ➤ Seleção

- Seleciona um par de indivíduos para reprodução
- Critério de seleção => função *fitness*
  
- 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.
- Tipo mais comum de seleção:
  - Seleção proporcional a aptidão

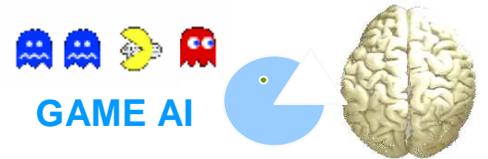
# Algoritmos Genéticos



## ➤ Reprodução

➤ Conjunto de operações (crossover, mutação, ...) aplicada a um par na geração de um novo indivíduo.

# Algoritmos Genéticos



## ➤ Crossover e Mutação

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

# Algoritmos Genéticos



## ➤ Mutaç o

- Pequenas mudanas aleat rias na seq ncia de bits
- Mutaç o inverte os valores dos bits.

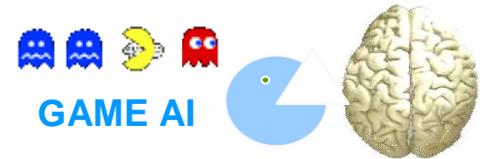
A mutaç o   aplicada com dada probabilidade, denominada *taxa de mutaç o* (~1%), em cada um dos bits do cromossomo.

A taxa de mutaç o n o deve ser nem alta nem baixa, mas o suficiente para assegurar a diversidade de cromossomos na populaç o.

**Antes da mutaç o** 0 1 1 0 1  
**Depois** 0 0 1 0 1

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

# Algoritmos Genéticos

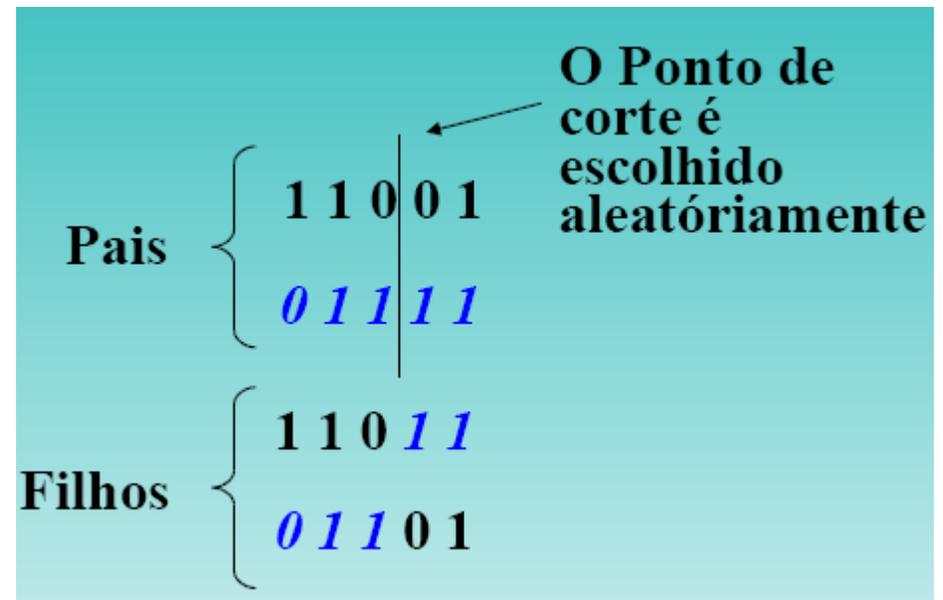


## ➤ Crossover

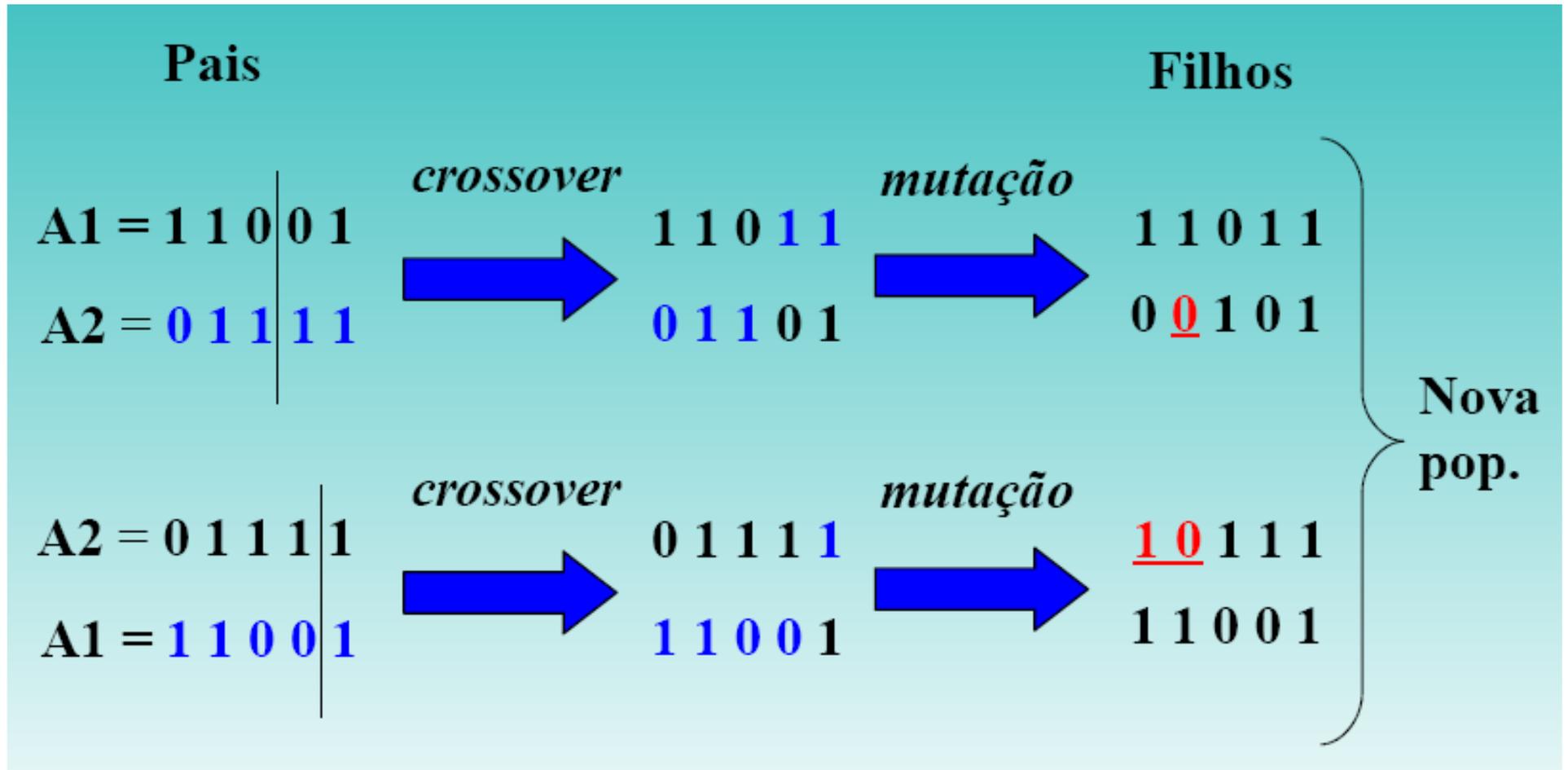
➤ Troca de material genético entre os pais.

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

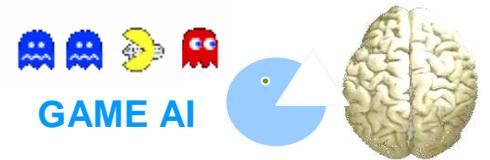
O crossover é aplicado com uma dada probabilidade denominada *taxa de crossover* (60% a 90%)



# Algoritmos Genéticos



# Algoritmos Genéticos



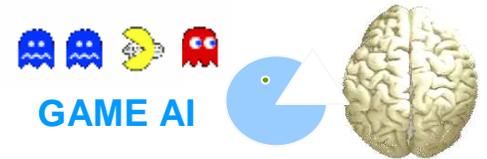
```
pai1 010011000101011
pai2 001001110001101
filho1 010001110101011
filho2 001011000001101
```

Crossover de 2 pontos

```
pai1 10101001001001
pai2 001001110001100
filho1 1010011100101011001
filho2 0010100100011001100
```

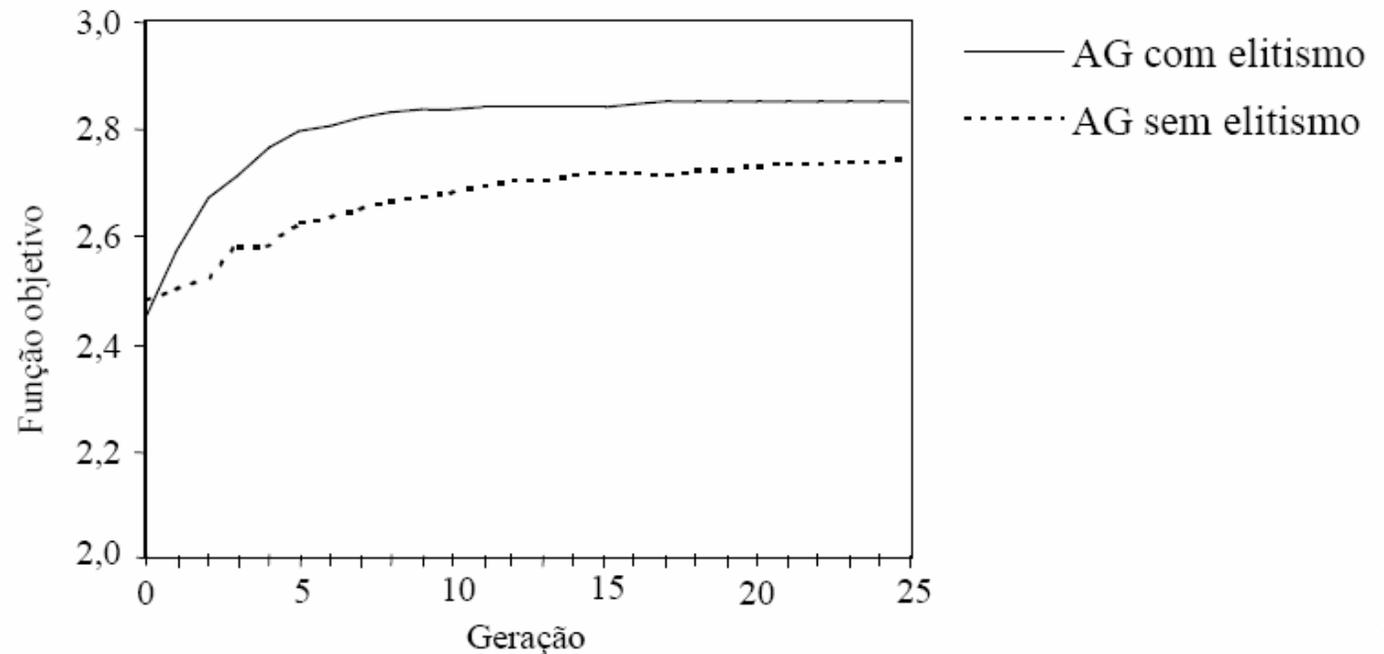
Crossover de 4 pontos

# Algoritmos Genéticos

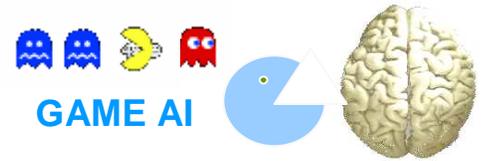


## ➤ Elitismo

- O melhor cromossomo pode ser perdido de uma geração para outra devido ao corte do crossover ou a ocorrência de mutação.
- Não é interessante transferir o melhor cromossomo de uma geração para outra sem alterações?
- Porque perder a melhor solução encontrada até então?
- Esta estratégia é denominada Elitismo



# Algoritmos Genéticos

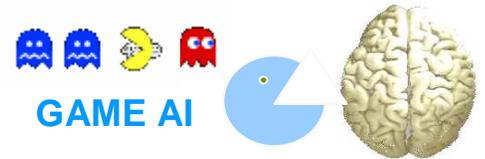


## ➤ Exemplo

➤ Achando o máximo de uma função

( AI4Games-GA-exemplo-funcao.ppt )

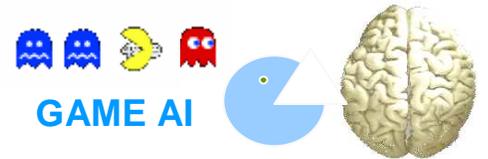
# Algoritmos Genéticos



## ➤ Fitness

- Function used to evaluate each member of one generation
- Measure individuals, obtaining a fitness value
- A função objetivo em alguns problemas pode ser bastante complicada, demandando um alto custo computacional. Por exemplo, existem problemas em que, para avaliar um cromossomo, é necessária uma simulação completa do processo, o que pode chegar a consumir horas.
- Haupt e Haupt (1998) sugerem, para lidar com tais funções objetivo, algumas dicas que propõem cuidados a serem tomados para não avaliar cromossomos idênticos mais de uma vez, reutilizando deste modo a avaliação feita anteriormente.
- Uma outra abordagem: procurar uma maneira de simplificar a função objetivo. A versão simplificada da função objetivo seria utilizada nas gerações iniciais para acelerar a busca por regiões promissoras do espaço de busca. Nas gerações finais, a versão original da função objetivo seria utilizada para melhorar a precisão da solução.

# Algoritmos Genéticos



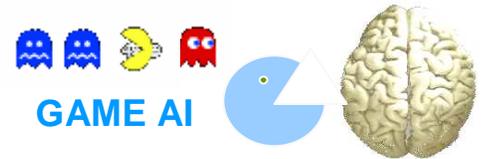
## ➤ Schema

### ➤ Propriedade:

➤ Set of bit strings that can be described by a template with wild-cards

➤  $1^{**}001 = 100001 / 101001 / 110001 / 111001$

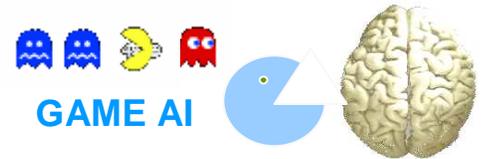
# Algoritmos Genéticos



## ➤ Critério de parada

- Número de gerações.
- Encontrou a solução (quando esta é conhecida).
- Convergência
  - nas últimas  $k$  gerações não houve melhora na aptidão:
    - Média
    - Máxima

# Algoritmos Genéticos



- Exemplo SELVA (`AI4Games-GA-exemplo-selva.ppt`)
- Exemplo Lunar Lander
- Exemplo PCV
- Exemplo Caminhar [Milton]
- Exemplo RoBombeiros

# Algoritmos Genéticos



**Mais material...**

**Obra classica: Goldberg, D. E. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, 1989.**

- **Tom M. Mitchell. Machine Learning. McGraw-Hill, 1997.**
- **Melanie Mitchell. An Introduction to Genetic Algorithms. MIT Press, Cambridge, 1999.**
- **Eberhart, R; Simpson, P; Dobbins, R. Computational Intelligence PC Tools. Academic Press, 1996.**
- **Lacerda, E. & Carvalho, A. Mini-Curso do JAI 1999 - Congresso da SBC. RJ, 1999.**
- **Holland 1975 / 1986, Davis 1991, De Jong 1993 => GA # Koza 1992, Fogel 1966, 1991, 1993 => GP**
- **Citeseer Nec: <http://www.researchindex.com/>**
- **FAQs: comp.ai.\* (Genetic, Alife, ...)**  
**<http://www.faqs.org/faqs/ai-faq/genetic/>**  
**<ftp://ftp.cerias.purdue.edu/pub/doc/EC/Welcome.html>**

**\* INTERNET:**

- **GA-GP.Com** <http://www.geneticprogramming.com/ga/index.htm>
- **GA Introduction** <http://islwww.epfl.ch/~moshes/ga.html>
- **GA Demo (Java)** <http://www.taygete.demon.co.uk/java/ga/index.html>
- **GA - TSP** <http://www.aridolan.com/ga/gaa/gaa.html>
- **GA Truck Demo** <http://www.handshake.de/user/blickle/Truck/index>