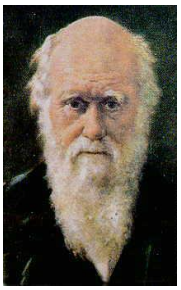
História da Teoria da Evolução



- **1859: Charles Darwin**
  - **Existe uma diversidade de seres devido aos contingentes da natureza (comida, clima, ...) e é pela lei da Seleção Natural que os seres mais adaptados ao seus ambientes sobrevivem**
    - contra lei do uso de desuso
  - **Os caracteres adquiridos são herdados pelas gerações seguintes**

"Quanto melhor um indivíduo se adaptar ao seu meio ambiente, maior será sua chance de sobreviver e gerar descendentes."

(DARWIN, 1859)

História da Teoria da Evolução

- **1865: Gregor Mendel**
  - **Formalizou a "herança de características", com a teoria do DNA (ervilhas)**
- **1901: Hugo De Vries**
  - **Só a seleção natural não é responsável pela produção de novas (mais adaptadas) espécies. Tem de haver uma mudança genética!**
  - **Formalizou o processo de geração de diversidade: Teoria da Mutação**

Surgimento dos Algoritmos Genéticos

- **1975: Jonh Holland: Idealizou os algoritmos genéticos**
- **Por que a evolução ?**
  - **Muitos problemas computacionais**
    - envolvem busca através de um grande número de possíveis soluções
    - requerem que o programa seja adaptativo, apto a agir em um ambiente dinâmico
  - **A evolução biológica**
    - uma busca paralela em um enorme espaço de problema
    - soluções desejadas = organismos mais adaptados

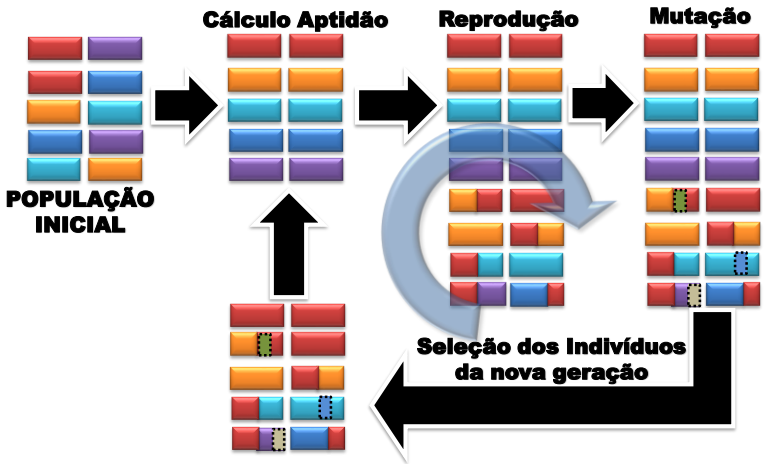
Algoritmos Genéticos: introdução

- **Algoritmos Evolutivos**
  - Método para resolução de problemas (otimização) "inspirado" na teoria da evolução
  - Algoritmos Genéticos são Algoritmos Evolutivos

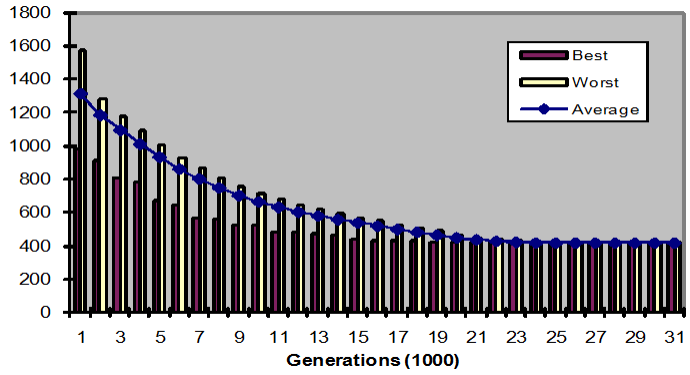
- **Algoritmo Genético**
  - indivíduo = solução
  - provoca mudança nos indivíduos por intermédio de *mutação* e *reprodução*
  - *seleciona* indivíduos mais adaptados através de sucessivas gerações
  - A aptidão de cada indivíduo é medida pela "função de aptidão" (*fitness function*)

Inteligência Artificial e Robótica - CC 7711
Algoritmos Genéticos

Ciclo do Algoritmo Genético



Convergência das gerações



Inteligência Artificial e Robótica - CC 7711
Algoritmos Genéticos

Gerando População Inicial

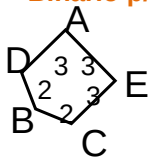
- **Dado um problema, precisamos definir como será a representação dos individuos (soluções)**
- **Um individuo é um Cromossomo**
  - **Os parâmetros do problema de otimização são representados por cadeias de valores.**
  - **Exemplos:**
    - Vetores de reais, (2.345, 4.3454, 5.1, 3.4)
    - Cadeias de bits, (111011011)
    - Vetores de inteiros, (1,4,2,5,2,8)
    - ou outra estrutura de dados (arvores, heaps, etc)

Gerando População Inicial

- **Qual deve ser a população Inicial ?**
  - **Deve ser aleatoriamente escolhida**
  - ***Precisamos garantir a variedade da população inicial:***
    - velocidade de convergência x variedade
    - Variedade x solução ótima
    - Quanto maior a variedade, menor a velocidade de convergência e vice-versa
    - Quanto maior a variedade, maior a chance de encontrar a solução ótima, porém, menor a velocidade de convergência

Função de Aptidão- Fitness

- **Função de aptidão (avaliação / fitness):**
  - É feita através de uma função que melhor representa o problema e tem por objetivo fornecer uma medida de aptidão de cada indivíduo na população corrente que irá dirigir o processo de busca.
  - Aptidão é uma nota associada ao indivíduo que avalia quão boa é a solução por ele representada
- **Funções de avaliação são específicas de cada problema.**
  - **Cromossomo** **Significado** **valor**
  - **0011011** **Binário p/ Inteiro** ***x* = 27**
  - **ADBCE**



$$\sum_{A \to D \to B \to C \to E} \text{dist} = 13$$

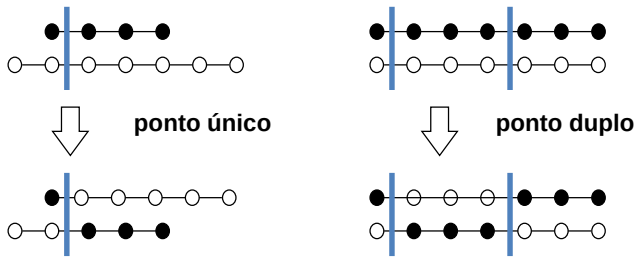
Criando e Evoluindo Gerações

- Quais são os métodos para criar novos individuos ?
  - Reprodução sexuada (crossover)
  - Reprodução assexuada
  - Mutação
- Como garantir encontrar a solução ?
  - Os indivíduos devem ser selecionados (os mais aptos sobrevivem) para garantir a convergência
  - Deve-se evitar a convergência prematura
  - Deve-se estabelecer um critério de parada
- Convergência
  - nas últimas *k* gerações não houve melhora da aptidão

Inteligência Artificial e Robótica - CC 7711
Algoritmos Genéticos

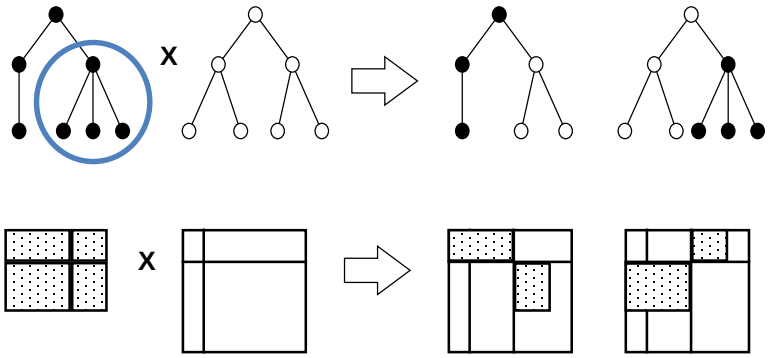
Reprodução/recombinação

- **Função:** combinar e/ou perpetuar material genético dos indivíduos mais adaptados
- **Tipos:**
  - assexuada (=duplicação) ou sexuada (crossover)
  - Observação: Existe uma "taxa de crossover" que controla a quantidade de reprodução que será feita



Reprodução

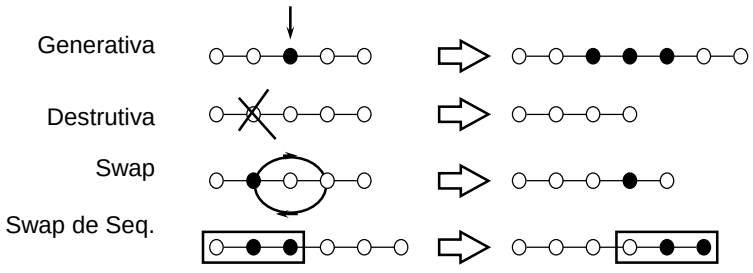
- Quanto mais "estruturada" a representação mais difícil de definir o cruzamento



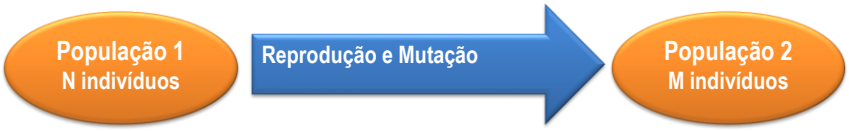
Inteligência Artificial e Robótica - CC 7711
Algoritmos Genéticos

Mutação

- **Objetivo: gerar diversidade (p/ escapar de ótimos locais)**
- **Tipos:**
  - generativa
  - destrutiva
  - swap
  - swap de seqüência
- **Observação: Existe uma "taxa de mutação" que diminui com o tempo (ex. % da população selecionada) para garantir convergência**



Reprodução e Mutação

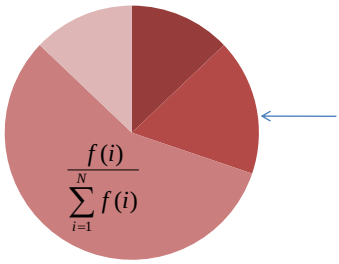


- **Composições possíveis para a População 2:**
  - **Troca de toda população**
    - Isso significa que a população 2 substitui a população 1
    - A cada ciclo, *N*/2 pares são escolhidos gerando *M*=*N* descendentes
  - **Elitismo**
    - A população 2 substitui a população 1 (M= N-1)
    - Acrescenta-se o mais apto da população 1 na população 2
  - **Steady State**
    - Gera-se M<N indivíduos e esses M substituem os M piores do conjunto da população 1 – Existe também o Steady state sem duplicados

Seleção de indivíduos para Reprodução

- **Como selecionar indivíduos para Reprodução ?**
- **Método da Roleta**
  - Selecionar indivíduos aleatoriamente, proporcionando chances de reprodução aos mais aptos.
  - ***f* (cromossomo) = medida numérica de aptidão**

Chances de seleção são proporcionais à aptidão



Método da Roleta

- **Encontre a soma da aptidão A_T de todos os indivíduos da população:**

$$A_T = \sum_{i=1}^{N} f(i)$$

- **Gere um número aleatório *r* :** $0 \le r \le A_T$
- **Selecione o primeiro indivíduo da população cuja aptidão somada às aptidões dos indivíduos precedentes é >= *r*:** $\sum_{i=1}^{k \le N} f(i) \ge r$

Exemplo: r = 30

Cromossomo	1	2	3	4	5	6	7	8	9	10
Aptidão	8	2	17	7	2	12	11	7	3	7
$\sum_{i=1} f(i)$	8	10	27	34	36	48	59	66	69	76

Critérios de parada

- **Definir um Número Máximo de gerações**
- **Parar quando o sistema encontrar a solução (quando esta é conhecida).**
- **Parar quando ocorrer a perda de diversidade (repetição individuos) → Convergir !!!**

Convergência Prematura

- **O que é a Convergência Prematura ?**
  - O sistema não apresenta mais nenhuma melhora e ainda está longe da situação ideal. Não apresenta mais diversidade da população
- **O que pode causar a Convergência Prematura ?**
  - Vários indivíduos iguais (pouca diversidade)
  - Um super-individuo que monopoliza o método da roleta
  - Uma baixa taxa de mutação e/ou reprodução
- **O que podemos fazer para evitar ?**
  - Mexer na função de aptidão (fitness) !!!!!
  - Existem técnicas que minimizam a criação de super-indivíduos e que aumentam a pressão seletiva sobre os melhores. São elas:
    - Transformação Exponencial
    - Normalização Linear (NL)

Evitando a Convergência Prematura

Transformação Exponencial:

- **O novo valor da Aptidão f' de cada individuo será dado por:**

$$f'(i) = \sqrt[2]{f(i)+1}$$

- **Reduz a influência do individuo mais forte**

Normalização Linear:

- **Ordene todos os *N* indivíduos em ordem decrescente de avaliação (*i* =1 menos apto)**
- **Crie aptidões partindo de um valor mínimo e crescendo linearmente até o valor máximo para cada um dos N indivíduos**

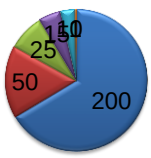
$$A_i = \min + \frac{(\max - \min)}{N-1} \times (i-1)$$

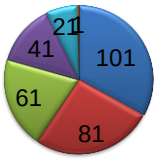
Idéia: Quanto menor a constante de incremento maior a pressão seletiva sobre os melhores. O valor normalmente usado é (max-min/N-1).

Comparativo

Exemplo Comparativo:

Rank	6	5	4	3	2	1
Aptidão	**200**	**50**	**25**	**15**	**10**	**2**
Transf. Exponencial	14	7	5	4	3	1
NL (incremento 20)	101	81	61	41	21	1





Algoritmos de AG mais usual

- **Possui as seguintes características:**
  - **Cromossomo:** Representação binária em um vetor
  - **Reprodução:** Crossover de 1 ponto
  - **Nova geração:** Reprodução (seleção) com substituição da população por Elitismo
  - **Convergência Prematura:** Normalização Linear
  - **Outros detalhes:** usa-se Mutação

Apresenta bom desempenho em diversas aplicações (é um bom algoritmo de partida).

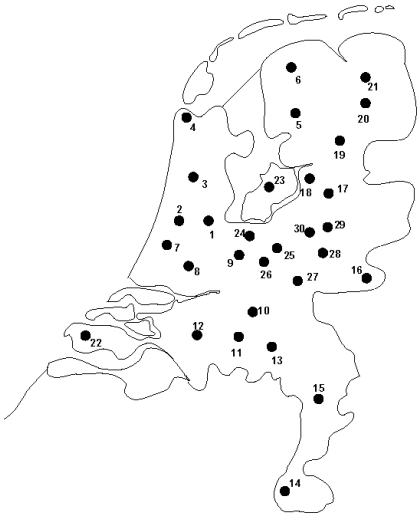
Um exemplo simples

O problema do Caixeiro Viajante:

Encontre um caminho entre as cidades de modo que:

- **Cada cidade seja visitada apenas uma vez**
- **A distância total de viagem seja minimizada**

- **Para um problema com 30 cidades, teremos algo como 30! (fatorial) possibilidades de caminhos**
- **$30! \approx 10^{32}$**



Representação

Iremos considerar um modo simples de representação:
- Lista ordenada de cidades visitadas
- Cada cidade será representada por um número

1) São Paulo	**3) SBCampo**	**5) Diadema**	**7) Guarulhos**
2) Campinas	**4) Sto André**	**6) Osasco**	**8) SCSul**

CityList1 (3 5 7 2 1 6 4 8)
CityList2 (2 5 7 6 8 1 3 4)

Definindo o Crossover

Iremos usar um Crossover de reprodução de ponto duplo

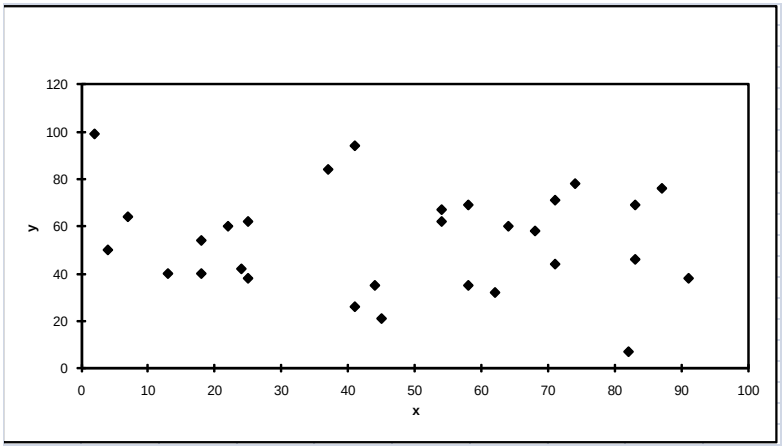
Pai1	(3	5	7	2	1	6	4	8)
Pai2	(2	5	7	6	8	1	3	4)
Filho	(2	5	7	2	1	6	3	4)

Definindo a Mutação

Mutação envolve reordenação de lista:

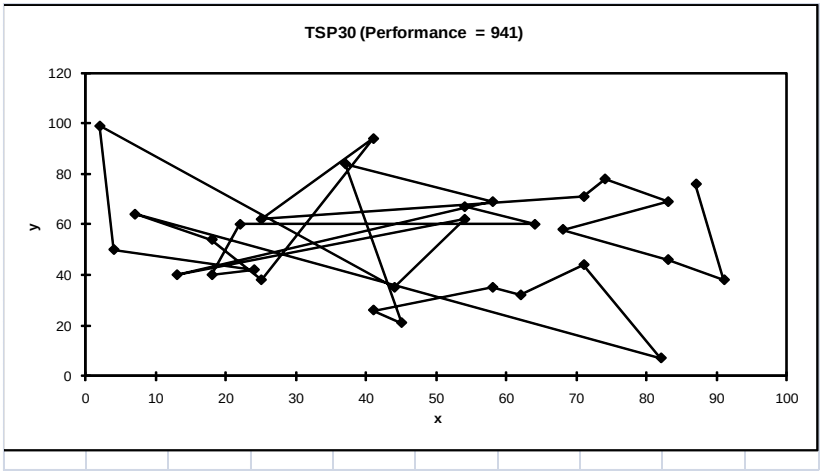
			*			*		
Antes:	(5	8	7	2	1	6	3	4)
Depois:	(5	8	6	2	1	7	3	4)

Exemplo: 30 Cidades

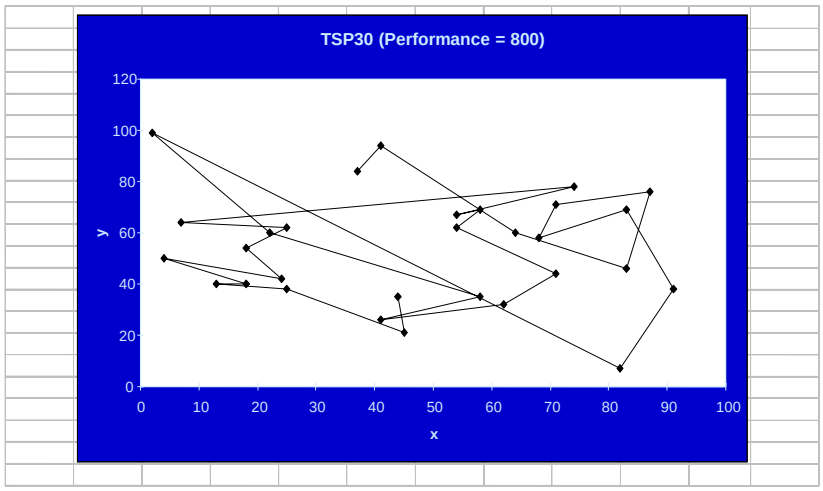


Inteligência Artificial e Robótica - CC 7711
Algoritmos Genéticos

Solução parcial (Distância = 941)

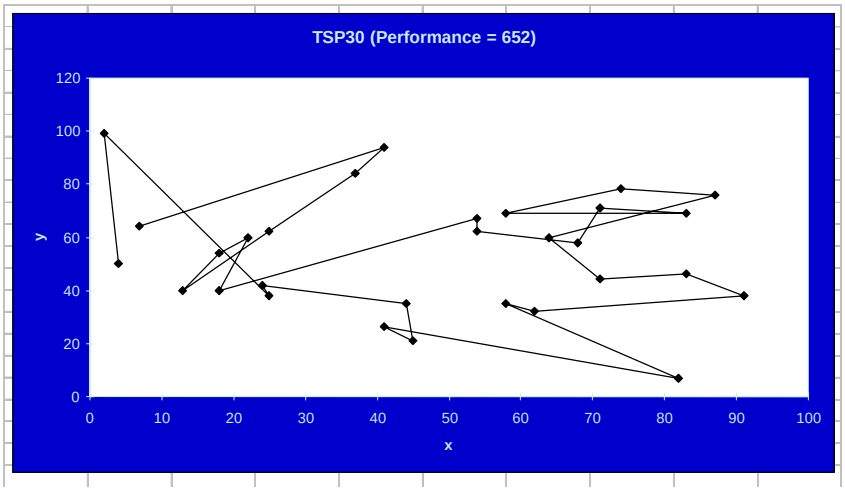


Solução parcial (Distância = 800)

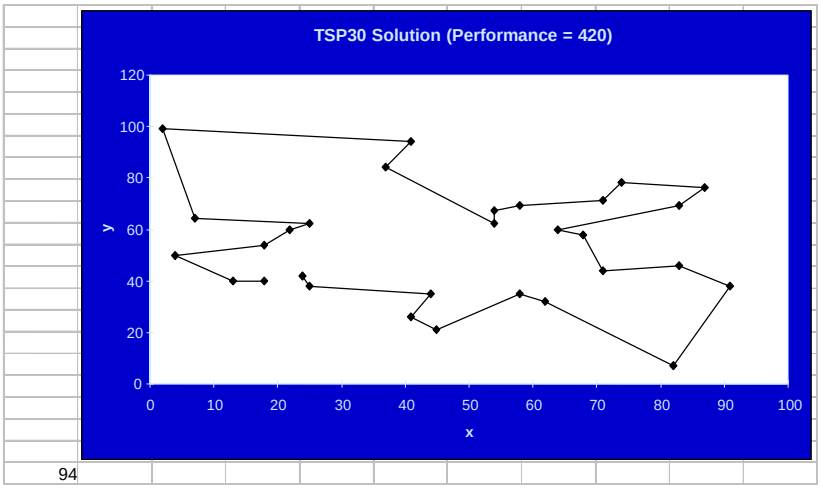


Inteligência Artificial e Robótica - CC 7711
Algoritmos Genéticos

Solução parcial (Distância = 652)

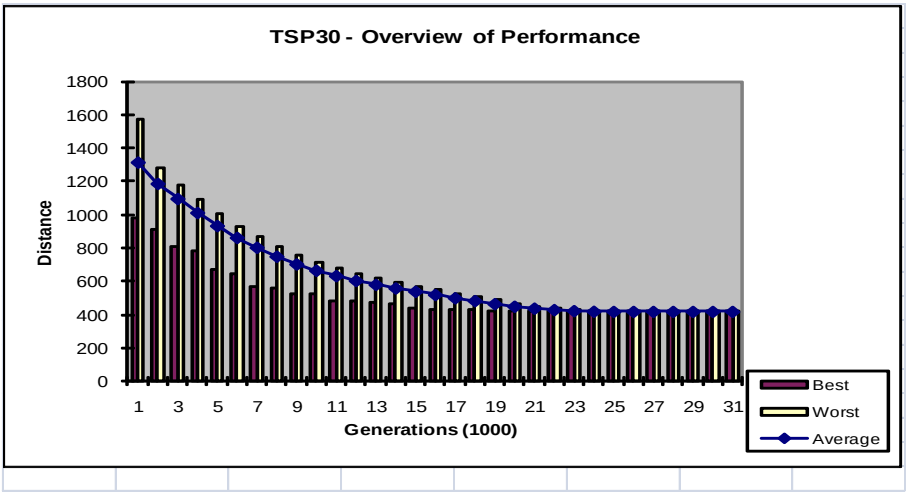


Solução parcial (Distância = 420)



Inteligência Artificial e Robótica - CC 7711
Algoritmos Genéticos

Gráfico da evolução do AG



Balanço

- Vantagens
  - Simples (várias representações, 1 algoritmo) e pouco sensível a pequenas variações
  - Vasto campo de aplicações
  - Ainda custa caro mas pode ser paralelizado facilmente

- Desvantagens
  - Como o método é basicamente numérico nem sempre é fácil introduzir conhecimento do domínio
  - Se não for paralelizado, pode demorar muito tempo para achar uma solução

Inteligência Artificial e Robótica - CC 7711
Algoritmos Genéticos

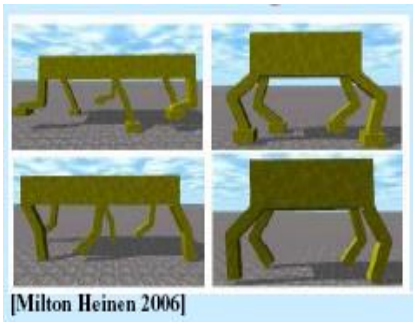
Aplicação prática (CMU robots)

http://www-math.uni-paderborn.de/~junge/images/aibos.jpg

- Fazer o AIBO andar mais rápido
- Aprendeu com Algoritmos Genéticos
- Deixaram alguns AIBOs andando de um lado para outro por dias.
- Eles andavam por x segundos e avaliavam o quanto haviam andado por posicionamento em campo
- Ele se localizava por LANDMARKS nos extremos do campo

Ensinar um quadrúpede andar

- Milton Heinen – Mestrado – 2007 - UNISINOS


[Milton Heinen 2006]

- Fitness: Ir mais longe, estável e mais rápido
- Ambiente e sensores Simulados.

Inteligência Artificial e Robótica - CC 7711
Algoritmos Genéticos

Bibliografia de Algoritmos Genéticos

aprofundamento nos assuntos desta aula, segue a seguinte referência bibliográfica

- Solange Rezende (Sistemas Inteligentes)
  - Capítulo 9

- Alguns slides desta aula foram baseados no slides:
- Geber Ramalho: "Algoritmos Evolutivos", UFPE.
- Wendy Willians. Genetic Algorithms: A tutorial.
- A.E. Eiben e J.E. Smith, Introduction to Evolutionary Computing Genetic Algorithms.
- Luis R. M. Lopes. Fundamentos dos Algoritmos Genéticos.
- Jennifer Pittman. Genetic Algorithm for Variable Selection. Duke University
- Carlos Eduardo Thomaz: Aulas do mestrado em RN. FEI 2010
- Grupo ICA – PUC-Rio