

Inteligência Computacional
COPPE - 2011

**COMPUTAÇÃO EVOLUTIVA &
Algoritmos Genéticos**

Alexandre P. Alves da Silva

Programa de Engenharia Elétrica

COPPE / UFRJ

alex@coep.ufrj.br

28/04/2011



Computação Natural

INTELIGÊNCIA COMPUTACIONAL

Computação Evolutiva

Redes Neurais
Artificiais

Lógica *Fuzzy*

Recozimento
Simulado

Sistemas
Complexos

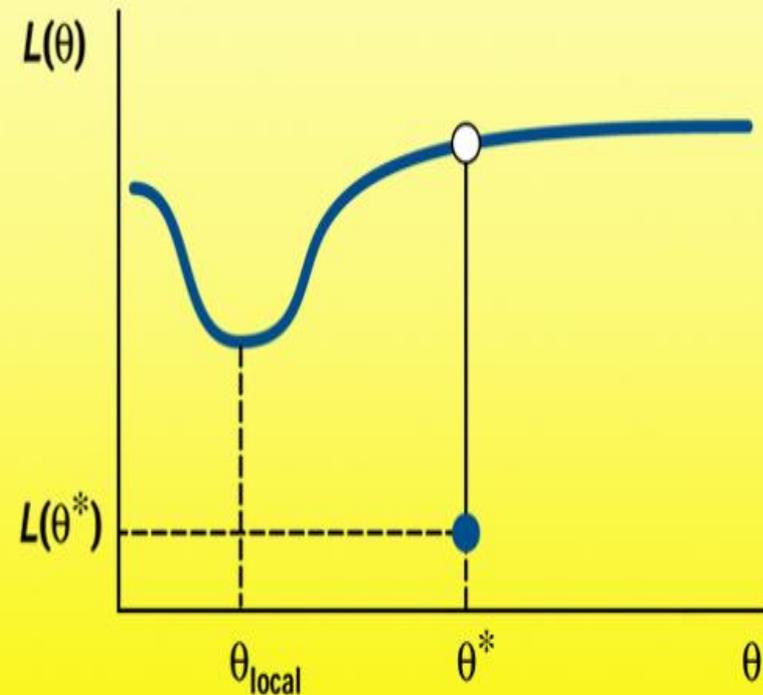
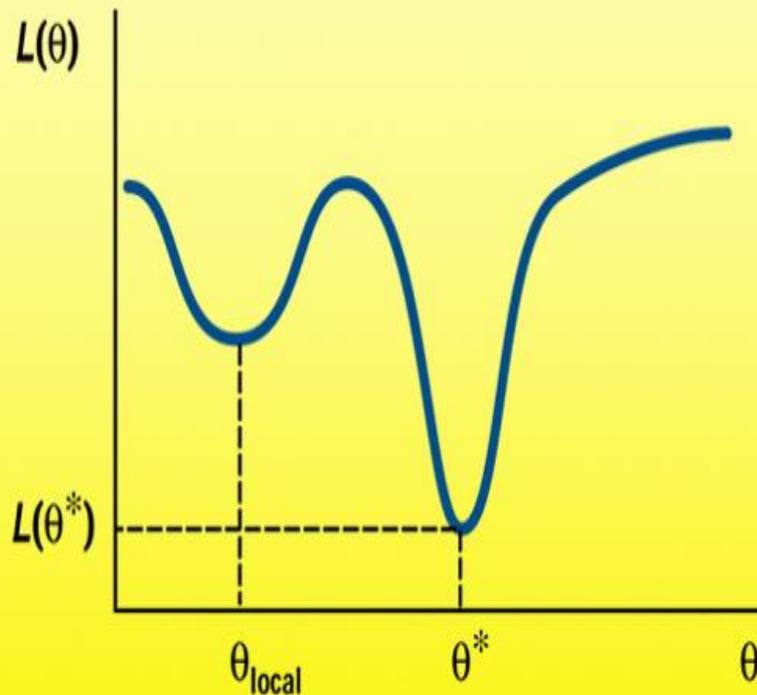
Vida
Artificial

Teoria do Caos

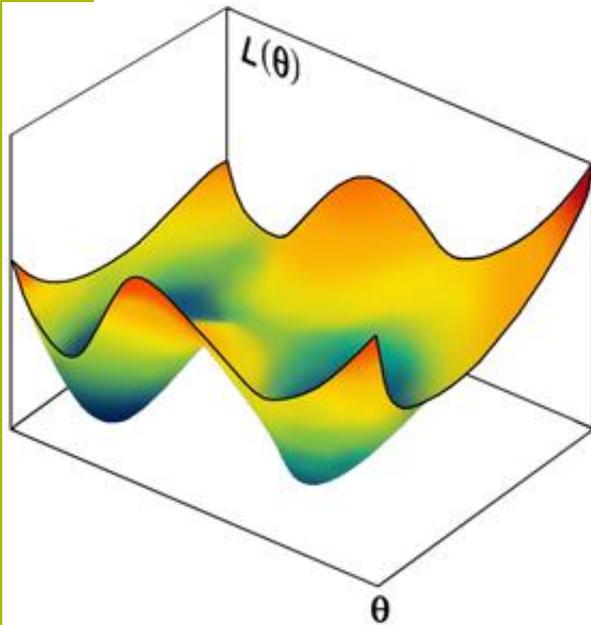
Algoritmos Evolutivos

Algoritmos Evolutivos são técnicas estocásticas de busca e otimização, **poderosas** e **largamente aplicáveis**, inspiradas nos mecanismos naturais da **evolução** e da **genética**

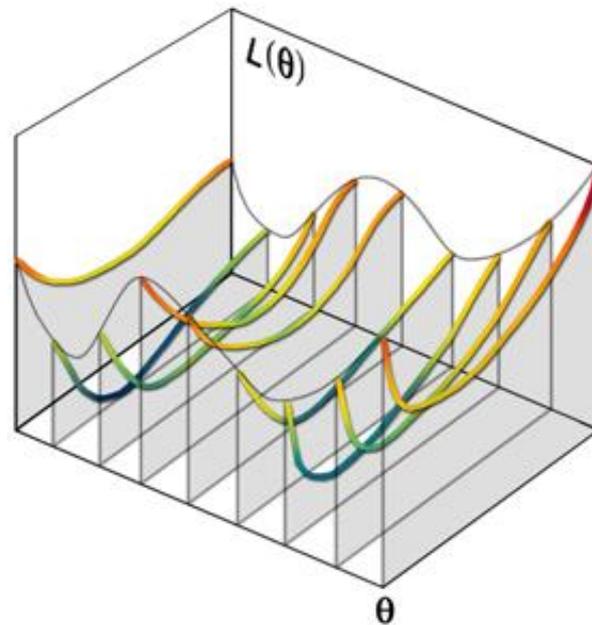
Exemplos de Problemas Fáceis e Difíceis em Otimização Global



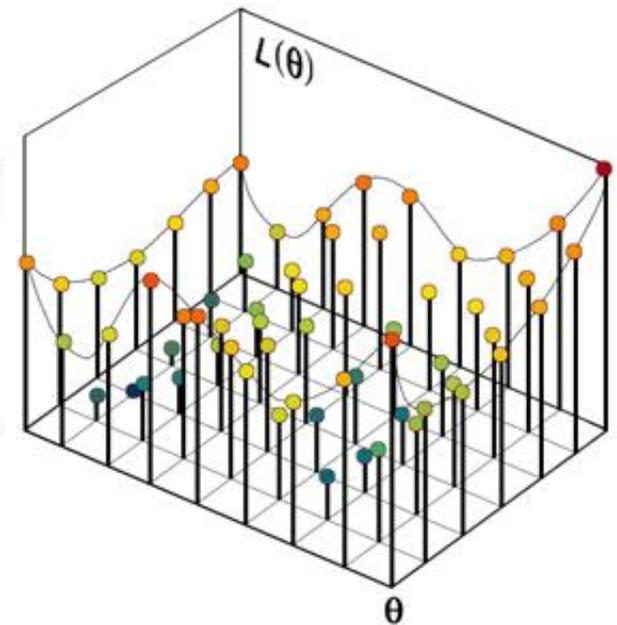
Três Tipos Comuns de Funções Objetivo



Contínua



Discreta/
Contínua



Discreta

Problemas de Engenharia

- Grandes dimensões
- Não-linearidades fortes
- Não-diferenciabilidade e/ou não-convexidade
- Funções não disponíveis ou não-tratáveis analiticamente (simulação, tabelas)
- Variáveis inteiras ou discretas (otimização combinatória)
- Multimodalidade (vários máximos e mínimos)

Técnicas de Otimização

Técnicas de Busca

Baseadas em Cálculo Diferencial

Fibonacci

"Sort"

Técnicas Estocásticas

Busca Tabu

Subida de Encosta

Recozimento Simulado

Algoritmos Evolutivos

Outros

Técnicas Enumerativas

DFS

Programação Dinâmica

BFS

Algoritmos Genéticos

Marcos Históricos

- Algoritmos Genéticos (AGs) (Holland, 1962)
- Programação Evolutiva (Fogel, 1962)
- Estratégias Evolutivas (Rechenberg & Schwefel, 1962)
- Programação Genética (Koza, 1990)
- “Particle Swarm Optimization” (Kennedy & Eberhart, 1995)
(inspiração em dinâmica comportamental)



MECANISMOS DE EXPLORAÇÃO DO ESPAÇO DE BUSCA

Busca sem Memória
(AG e RS)

X

Memória Adaptativa
(BT)

Operadores Estocásticos
(AG e RS)

X

Exploração Sistemática de Vizinhança
(BT)

Transições Pontuais
(RS e BT)

X

Baseadas em Populações
(AG)

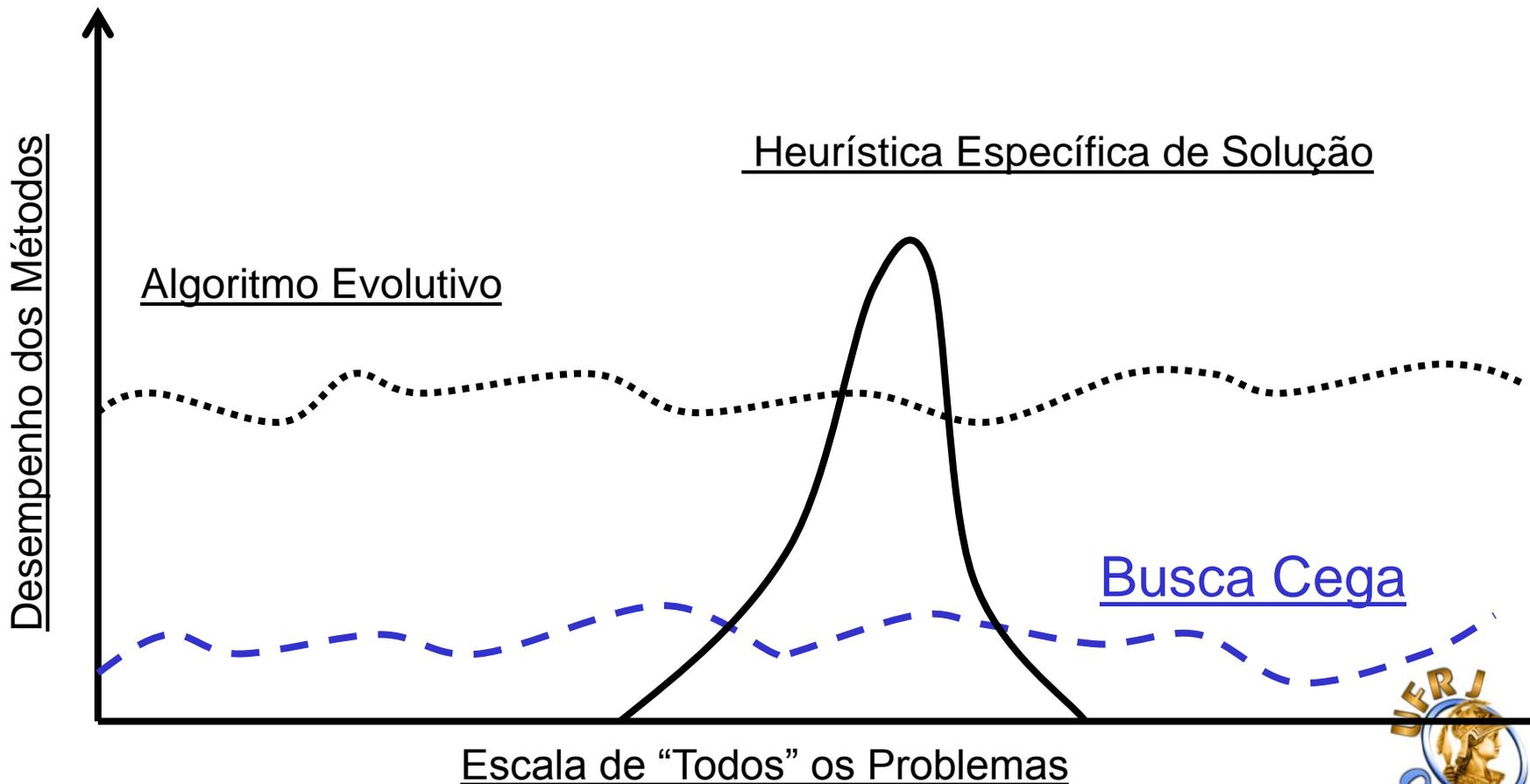


Diversidade da Busca
("global exploration")
Qualidade dos AGs

&

Capacidade de Intensificação
("local exploitation")
Deficiência dos AGs

Relativizando a Computação Evolutiva



Metáfora Biológica

- **População:** conjunto de estruturas (no caso de AG, PE e EE são soluções potenciais)
- **Indivíduos:** cada uma das estruturas
- **Geração:** cada passo do processo evolutivo
- **Princípios:**
 - **Evolução:** Sobrevivência do mais apto
 - **Genética:** recombinação e mutação

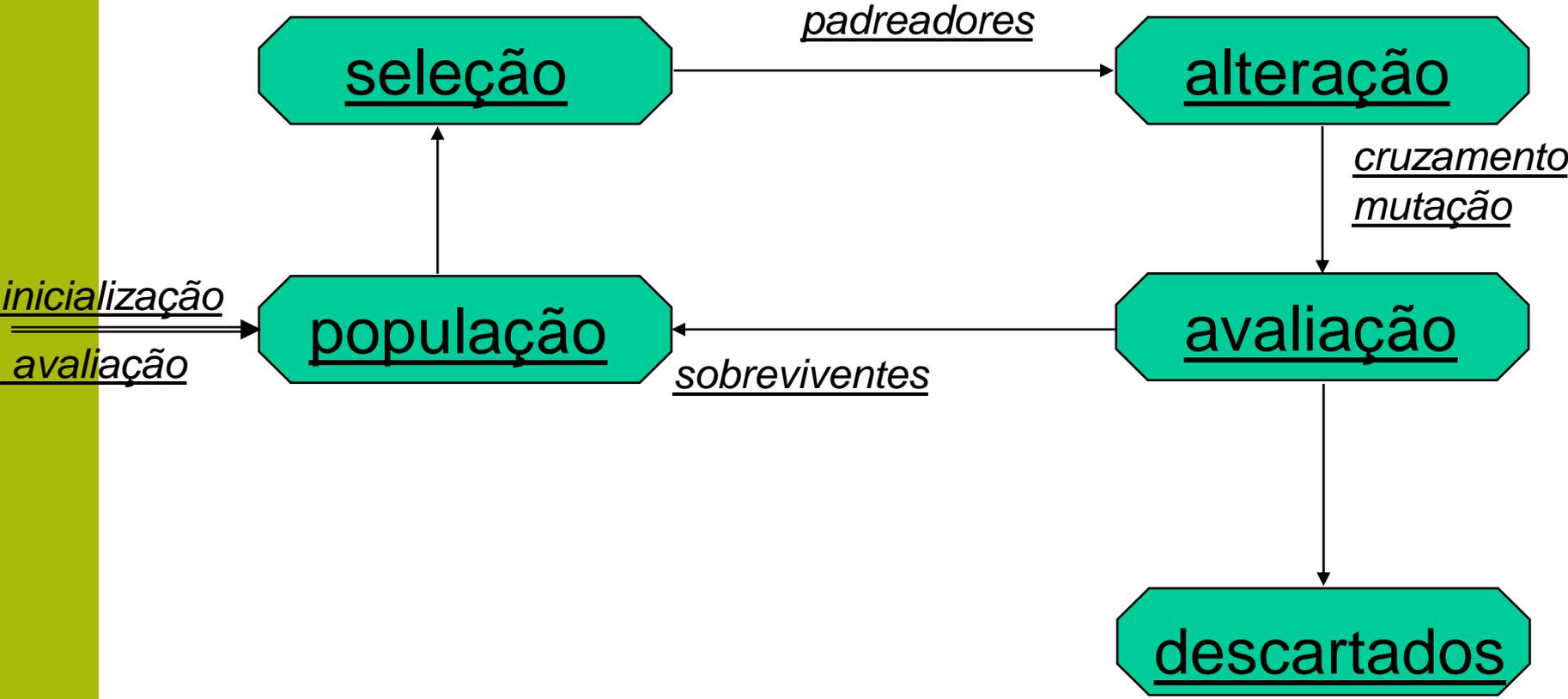
Princípio Básico

Uma população de **estruturas computacionais** evolui de forma tal que existe uma melhoria na **adequação média** ao ambiente dos indivíduos dessa população

Elementos do Algoritmo

- Inicialização: aleatória (em geral)
- Avaliação: função adequabilidade (*fitness function*)
- Seleção: escolhe *melhores* indivíduos
- Alteração: operadores genéticos (cruzamento e mutação)
- Critério de parada: estagnação, objetivo alcançado, tempo, número de gerações

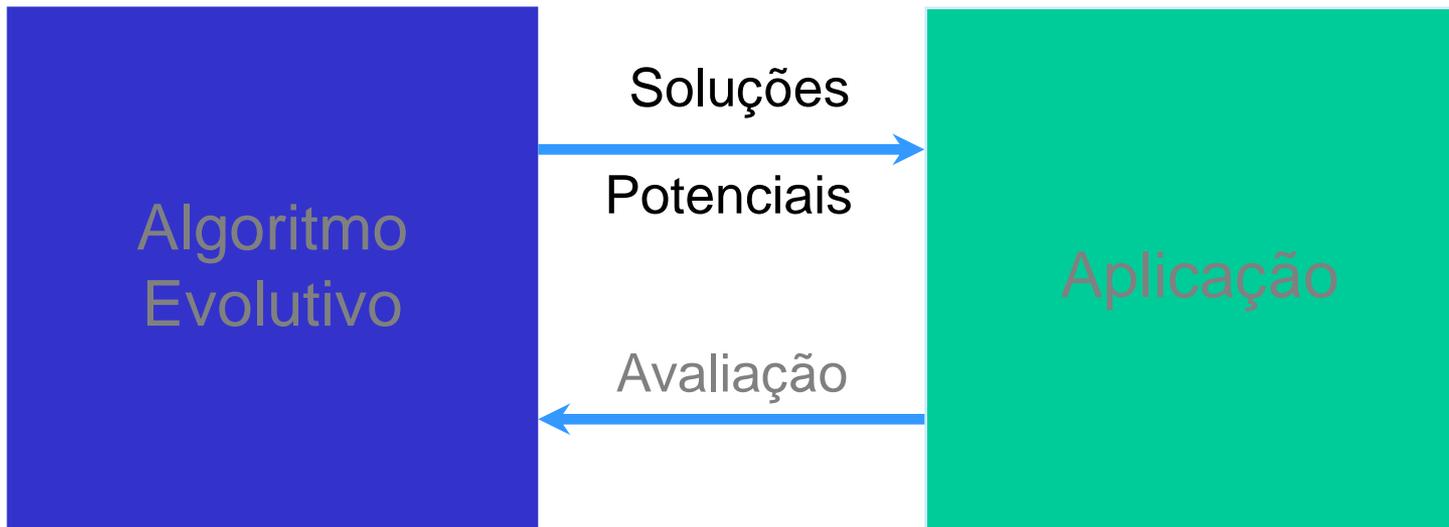
Ciclo Evolutivo



Aplicações

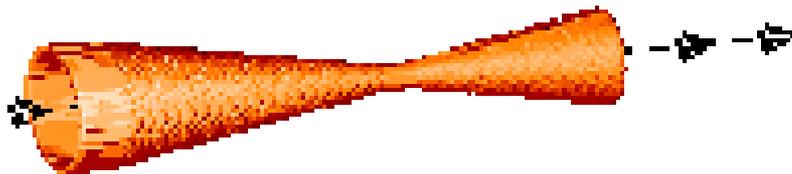
- Otimização
 - Projetos
 - Tomada de decisões
 - Controle
 - Etc.
- Síntese e treinamento de redes neurais
- Regras de inferência *fuzzy*
- Aprendizado de máquina (*machine learning*)

Interação com Aplicações

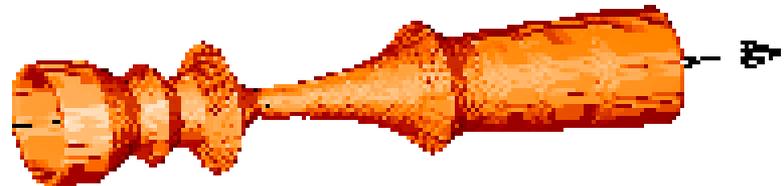


Aplicação Prática dos Anos 60: projeto de turbina a jato

Tarefa: otimizar a geometria de turbinas a jato
Abordagem de solução: seleção + mutação



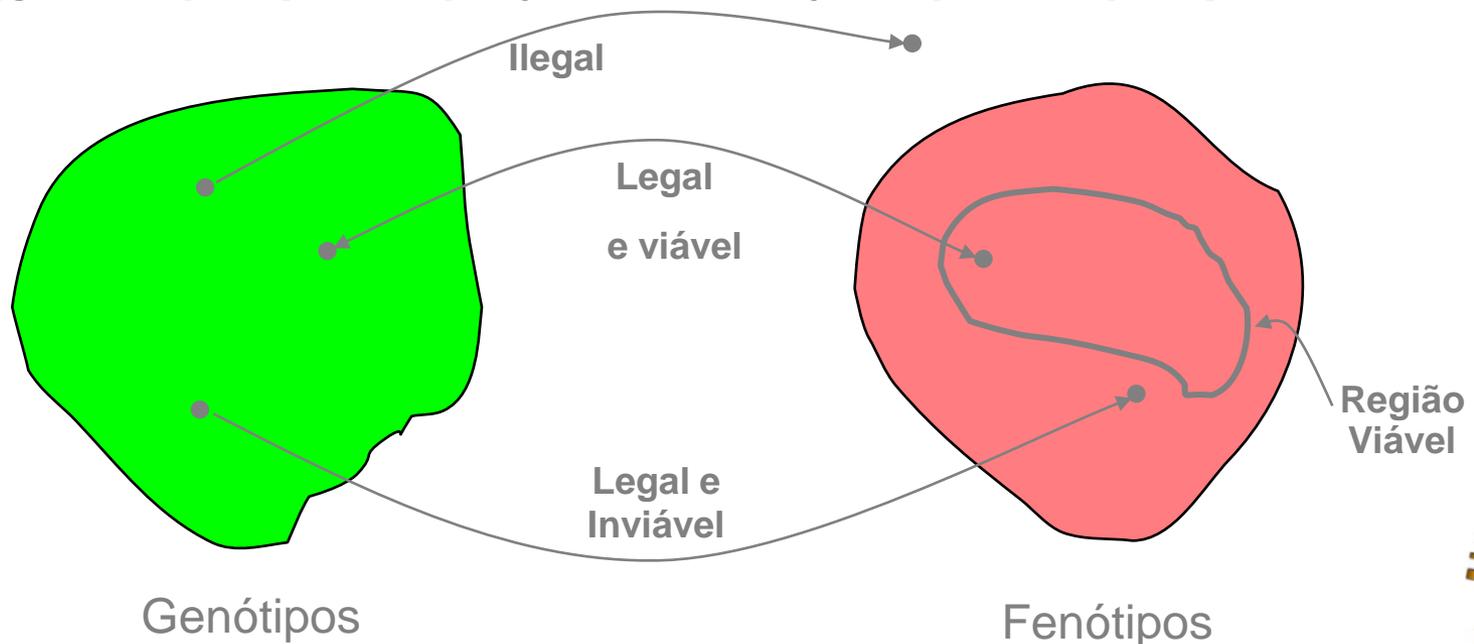
Formato Inicial



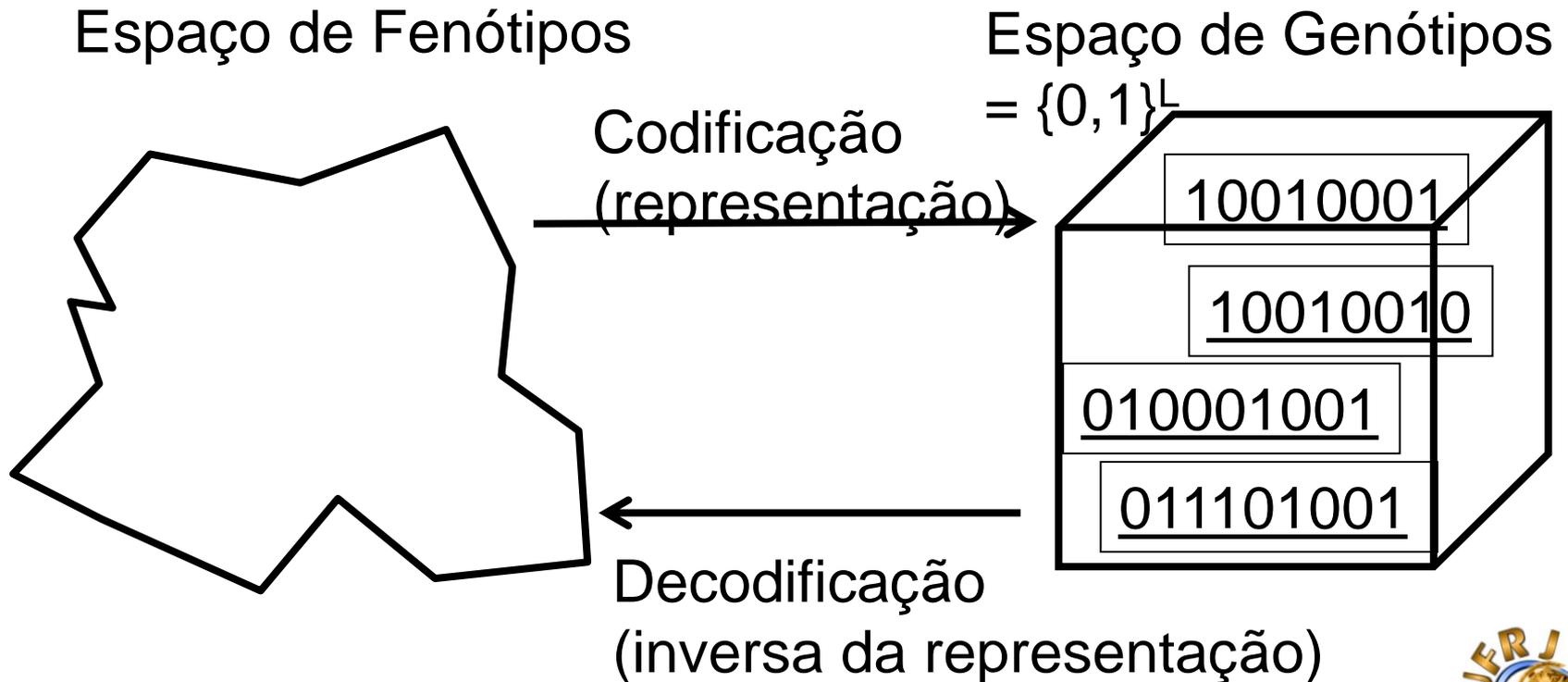
Formato Final

Codificação/Decodificação em AGs

- Forma compacta de representação de soluções potenciais do problema
- Relação bi-unívoca entre espaço de codificação (genótipos) e espaço de solução (fenótipos)



Representação Binária

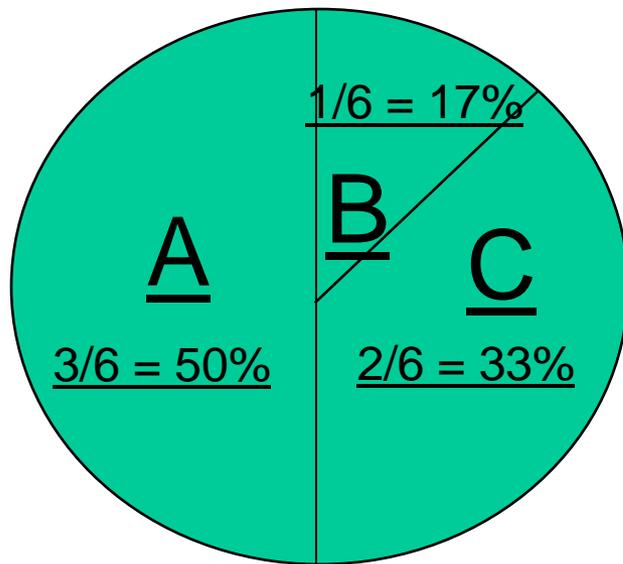


Ciclo do AG "Simples"

1. Selecionar padreadores
(número de padreadores = tamanho da população)
2. Misturar os padreadores
3. Para cada par de padreadores aplicar o operador cruzamento com probabilidade p_c , caso contrário reproduzir o par de padreadores
4. Para cada cria aplicar mutação ("bit-flip") com probabilidade p_m , de forma independente para cada bit
5. Gerar uma nova população com as crias resultantes

Seleção Proporcional

- Idéia: melhores indivíduos têm mais possibilidades de reprodução
 - Chance proporcional à adequabilidade
 - Implementação: roleta



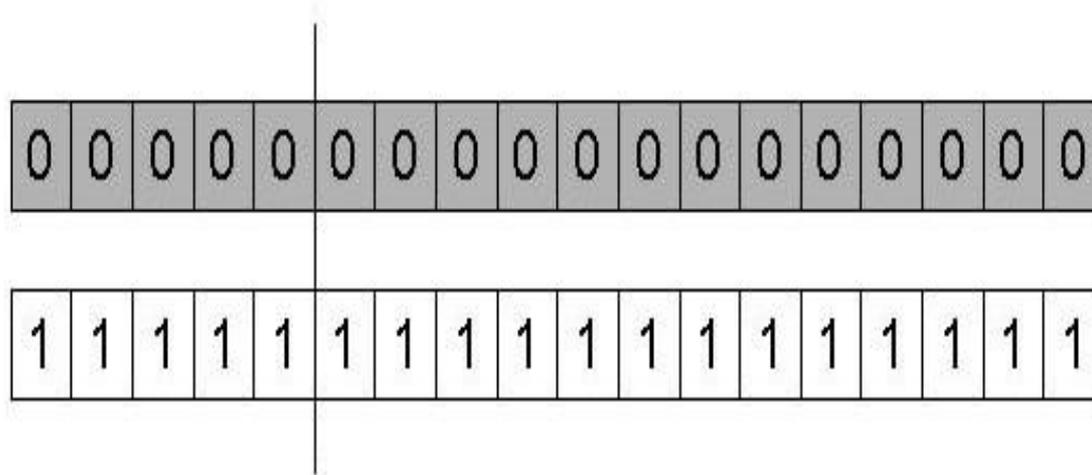
adequabilidade de (A) = 3

← adequabilidade de (B) = 1

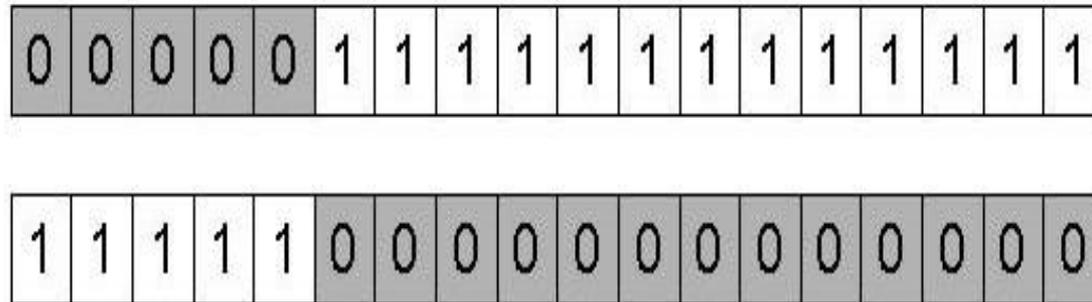
adequabilidade de (C) = 2

Cruzamento em 1 ponto

parents



children



Mutação ("bit-flip")

parent

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

child

0	1	0	0	1	0	1	1	0	0	0	1	0	1	1	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Exemplo [Goldberg, 1989]

- Problema: $\max x^2$ no domínio $\{0,1,\dots,31\}$
- Abordagem de Algoritmo Genéticos:
 - Representação: código binário, ex., 01101 \leftrightarrow 13
 - Tamanho da população: 4
 - Seleção proporcional (roleta)
 - Cruzamento em 1 ponto
 - Mutação com base em troca de bit
 - Inicialização aleatória

Exemplo x^2 : seleção

String no.	Initial population	x Value	Fitness $f(x) = x^2$	$Prob_i$	Expected count	Actual count
1	0 1 1 0 1	13	169	0.14	0.58	1
2	1 1 0 0 0	24	576	0.49	1.97	2
3	0 1 0 0 0	8	64	0.06	0.22	0
4	1 0 0 1 1	19	361	0.31	1.23	1
Sum			1170	1.00	4.00	4
Average			293	0.25	1.00	1
Max			576	0.49	1.97	2

Exemplo x^2 : cruzamento

String no.	Mating pool	Crossover point	Offspring after xover	x Value	Fitness $f(x) = x^2$
1	0 1 1 0 1	4	0 1 1 0 0	12	144
2	1 1 0 0 0	4	1 1 0 0 1	25	625
2	1 1 0 0 0	2	1 1 0 1 1	27	729
4	1 0 0 1 1	2	1 0 0 0 0	16	256
Sum					1754
Average					439
Max					729

Exemplo x^2 : mutação

String no.	Offspring after xover	Offspring after mutation	x Value	Fitness $f(x) = x^2$
1	0 1 1 0 0	1 1 1 0 0	26	676
2	1 1 0 0 1	1 1 0 0 1	25	625
2	1 1 0 1 1	1 1 0 1 1	27	729
4	1 0 0 0 0	1 0 1 0 0	18	324
Sum				2354
Average				588.5
Max				729

Por que os AGs funcionam?

- schema → todos os possíveis strings com os mesmos bits em certas posições
ex., $**000 \rightarrow \{00000, 01000, 10000, 11000\}$
- adequabilidade média de um schema
- comprimento do esquema – “defining length” (DL)
ex., $**011* \rightarrow DL=2$
 $1*0**1 \rightarrow DL=5$
- “building block” → schema com alta adequabilidade e DL pequeno

CONCLUSÃO:

[00000001 | 00000001]

INTELIGÊNCIA BELEZA

[11111111 | 11111111]

INTELIGÊNCIA BELEZA



DESASTRE



ÓTIMO GLOBAL!!!

Algoritmo Genético "Simple"

- Foi motivo de diversos estudos
 - Ainda, eventualmente, utilizado como "benchmark"
- Apresenta muitas limitações:
 - Representação é muito restritiva
 - Operadores de cruzamento e mutação aplicáveis somente para representações binárias / discretas
 - Mecanismo de seleção induz à convergência prematura
 - Modelo de geração de populações pode ser melhorado com base na utilização de mecanismo de seleção de sobreviventes

Referências Selecionadas

- Fogel, D. B. (2000), *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence* (2nd ed.), IEEE Press, Piscataway, NJ.
- Fu, M. C. (2002), "Optimization for Simulation: Theory vs. Practice" (with discussion by S. Andradóttir, P. Glynn, and J. P. Kelly), *INFORMS Journal on Computing*, vol. 14, pp. 192–227.
- Goldberg, D. E. (1989), *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA.
- Gosavi, A. (2003), *Simulation-Based Optimization: Parametric Optimization Techniques and Reinforcement Learning*, Kluwer, Boston.
- Holland, J. H. (1975), *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI.
- Kushner, H. J. and Yin, G. G. (2003), *Stochastic Approximation and Recursive Algorithms and Applications* (2nd ed.), Springer-Verlag, New York.
- Michalewicz, Z. and Fogel, D. B. (2000), *How to Solve It: Modern Heuristics*, Springer-Verlag, New York.
- Spall, J. C. (2003), *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*, Wiley, Hoboken, NJ.
- Zhigljavsky, A. A. (1991), *Theory of Global Random Search*, Kluwer Academic, Boston.