# 1. Computação Evolutiva

**Prof. Renato Tinós**

Programa de Pós-Graduação Em Computação Aplicada

Depto. de Computação e Matemática (FFCLRP/USP)

# 1.3. Estratégias Evolutivas

1.3.1. Introdução às Estratégias Evolutivas
1.3.2. Seleção
1.3.3. Mutação
1.3.4. Codificação
1.3.5. Recombinação
1.3.6. Auto-adaptação
1.3.7. Exemplos

# 1.3.1. Introdução às Estratégias Evolutivas

- Estratégia Evolutiva (Evolution Strategy – ES)
  - Proposta na Alemanha, principalmente por I. Rechenberg, H.-P. Schwefel
    - nas décadas de 1960 e 1970
  - Tipicamente utilizada em otimização de soluções compostas por elementos que assumem valores **contínuos** (reais)
  - Características
    - Em geral rápida
    - Existem várias arquiteturas
    - Teoria bem desenvolvida (quando comparada a outros Algoritmos Evolutivos)

# 1.3.1. Introdução às Estratégias Evolutivas

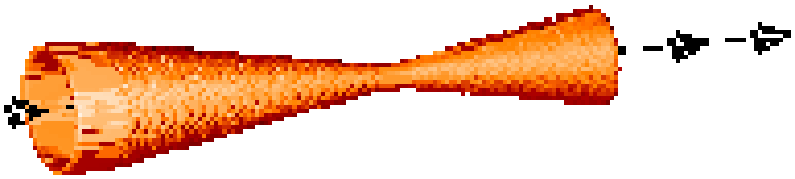
Sumário: características geralmente encontradas

| | |
|---|---|
| Representação | Vetores com valores reais |
| Recombinação | Discreta ou intermediária |
| Mutação | Gaussiana com possível auto-adaptação dos parâmetros de mutação |
| Seleção dos pais | Aleatória com distribuição uniforme |
| Seleção para próxima população | $(\mu,\lambda)$ ou $(\mu+\lambda)$ |

# 1.3.1. Introdução às Estratégias Evolutivas

Exemplo 1.3.1. *Jet nozzle experiment*

- Tarefa: otimizar o formato de um bocal de duas fases
- Método: mutações aleatórias, seleção($\mu$+$\lambda$)



http://ls11-www.cs.uni-dortmund.de/people/schwefel/EADemos/#bot

# 1.3.1. Introdução às Estratégias Evolutivas

Exemplo 1.3.1.

- Tarefa:

$$\begin{array}{ll} \text{minimize} & f(\mathbf{x}) \in \mathrm{R} \\ \text{sujeito a} & \mathbf{x} \in \mathrm{R}^n \end{array}$$

- Algoritmo: ES com um único indivíduo
  - Cromossomos: vetores reais *n*-dimensionais
  - Tamanho da população: 1
  - Mutação cria um filho a partir do único pai
  - Seleção elitista
  - Observação: no trabalho original, algumas restrições tiveram que ser feitas

# 1.3.1. Introdução às Estratégias Evolutivas

Exemplo 1.3.1.

**Algoritmo ES com um único indivíduo**

**Início**

$t \leftarrow 0$

Crie ponto inicial $\mathbf{x}_t = [\, x_t(1), \ldots, x_t(n) \,]^T$

avalie indivíduo $\mathbf{x}_t$

**enquanto** ( criterioConvergencia == 0 )

gere $\mathbf{z}_t$ a partir de distribuição normal

$\mathbf{y}_t \leftarrow \mathbf{x}_t + \mathbf{z}_t$

avalie indivíduo $\mathbf{y}_t$

**se** $f(\mathbf{x}_t) < f(\mathbf{y}_t)$

$\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t$

**senão**

$\mathbf{x}_{t+1} \leftarrow \mathbf{y}_t$

**fim se**

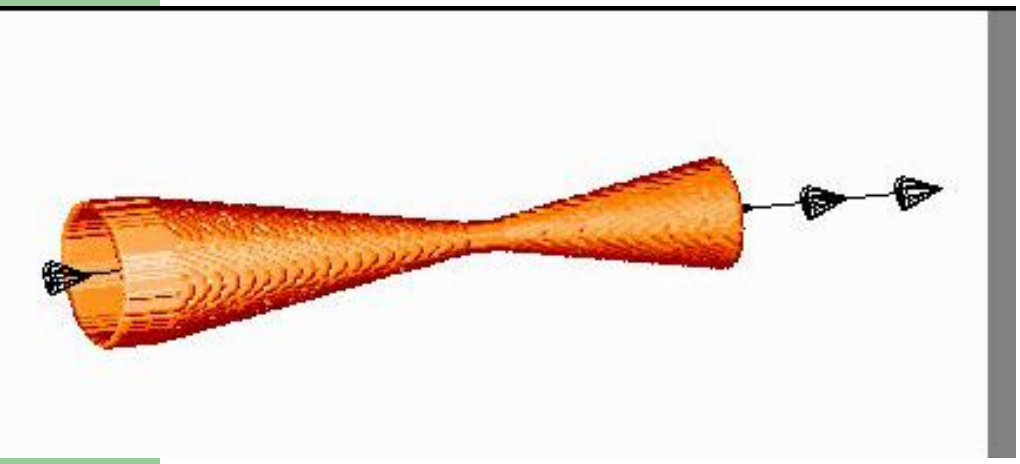
$t \leftarrow t + 1$

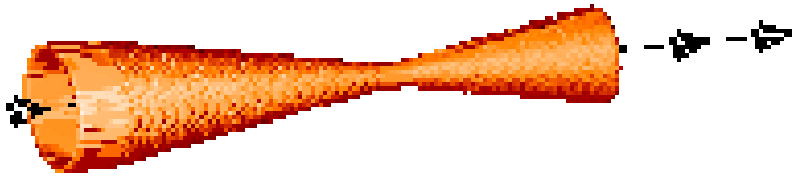
**fim enquanto**

**Fim**

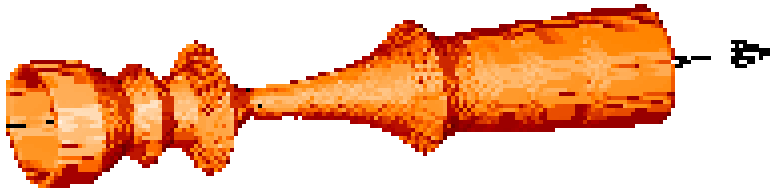
# 1.3.1. Introdução às Estratégias Evolutivas

Exemplo 1.3.1.





Initial shape



Final shape

# 1.3.2. Seleção

**Algoritmo Evolutivo Estacionário**

**Início**

$t \leftarrow 1$

inicializePopulacao( $\mathbf{P}_t$ )

avaliePopulacao( $\mathbf{P}_t$ )

**enquanto** ( criterioConvergencia == 0 )

$\mathbf{Q}_t \leftarrow$ transformePopulacao( $\mathbf{P}_t$ )

avaliePopulacao( $\mathbf{Q}_t$ )

$\mathbf{P}_{t+1} \leftarrow$ selecao( $\mathbf{P}_t \cup \mathbf{Q}_t$ )

$t \leftarrow t + 1$

**fim enquanto**

**Fim**

# 1.3.2. Seleção

- **( $\mu$ + $\lambda$ )-ES**
  - Uma população com $\mu$ pais cria $\lambda$ descendentes utilizando mutação e recombinação
  - Os melhores $\mu$ indivíduos entre os $\mu$ pais e os $\lambda$ descendentes formam a nova população
    - Elitista
  - Algoritmo Evolutivo Estacionário

# 1.3.2. Seleção

- **( $\mu$ , $\lambda$ )-ES**
  - Uma população com $\mu$ pais cria $\lambda \geq \mu$ descendentes utilizando mutação e recombinação
  - Os melhores $\mu$ descendentes formam a nova população
    - Não é elitista
  - Variação:
    - $\mathbf{P}_{t+1} \leftarrow$ selecao( $\mathbf{Q}_t$ )

# 1.3.2. Seleção

- ( $\mu$ , $\lambda$ )-ES ou ( $\mu$ + $\lambda$ )-ES ?
  - Muitas vezes ( $\mu$ , $\lambda$ )-ES é utilizada pois, geralmente,
    - É melhor em escapar de ótimos locais
    - É melhor para seguir ótimos que se movem continuamente
    - Não conserva (como a ( $\mu$ + $\lambda$ )-ES ) por muito tempo valores ruins dos parâmetros que geram o passo de mutação (quando estes são evoluídos) em indívíduos promissores
- Pressão seletiva geralmente é mantida alta em ESs
  - $\lambda \approx 7 \cdot \mu$ é muitas vezes utilizada

# 1.3.2. Seleção

- Seleção dos indivíduos a serem reproduzidos
  - Pais são selecionados aleatoriamente com distribuição uniforme
    - para qualquer tipo de operador de reprodução
    - assim, a seleção é sem bias
      - todo indivíduo tem a mesma chance de ser selecionado para ser reproduzido

# 1.3.3. Mutação

- Geralmente é utilizada a mutação Gaussiana

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \mathbf{C}_{t+1}\,\mathbf{N}(0,1)$$

**N**(0,1) é um vetor aleatório gerado a partir de uma distribuição normal com média 0 e variância 1

A matriz **C** define o espalhamento da distribuição

- Outras distribuições podem ser utilizadas
  - Exemplo: Cauchy

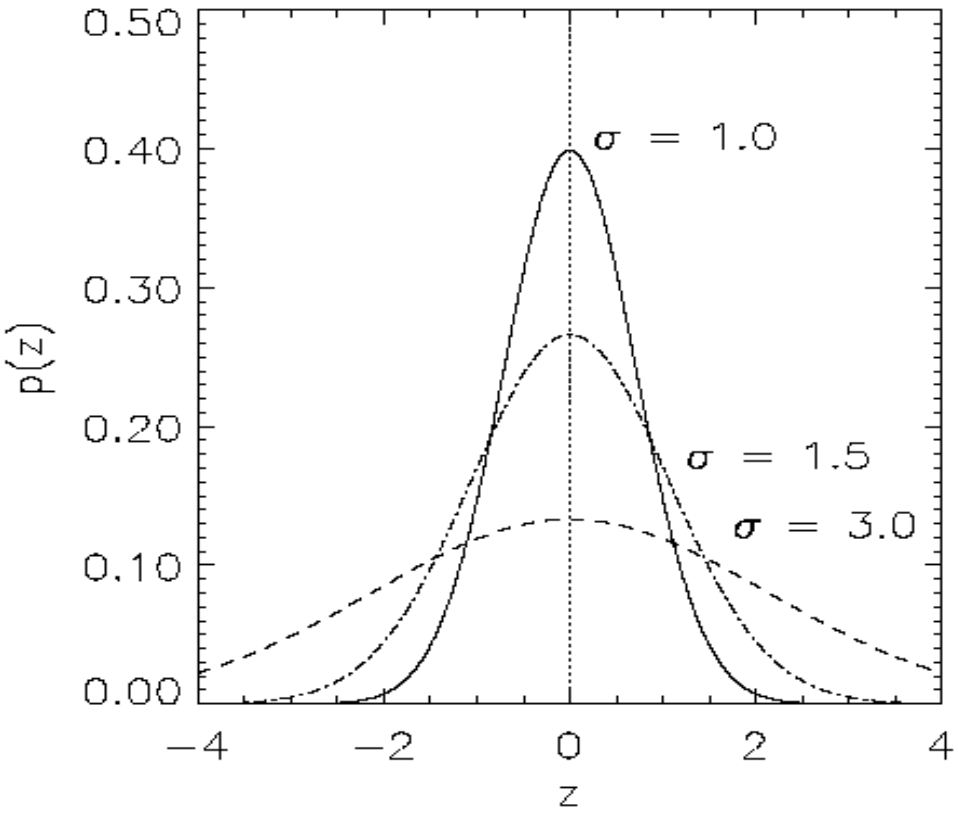
# 1.3.3. Mutação

- Mutação Gaussiana Não-correlacionada com um parâmetro de mutação
  - Matriz **C** definida por

$$\mathbf{C} = \sigma\,\mathbf{I}$$

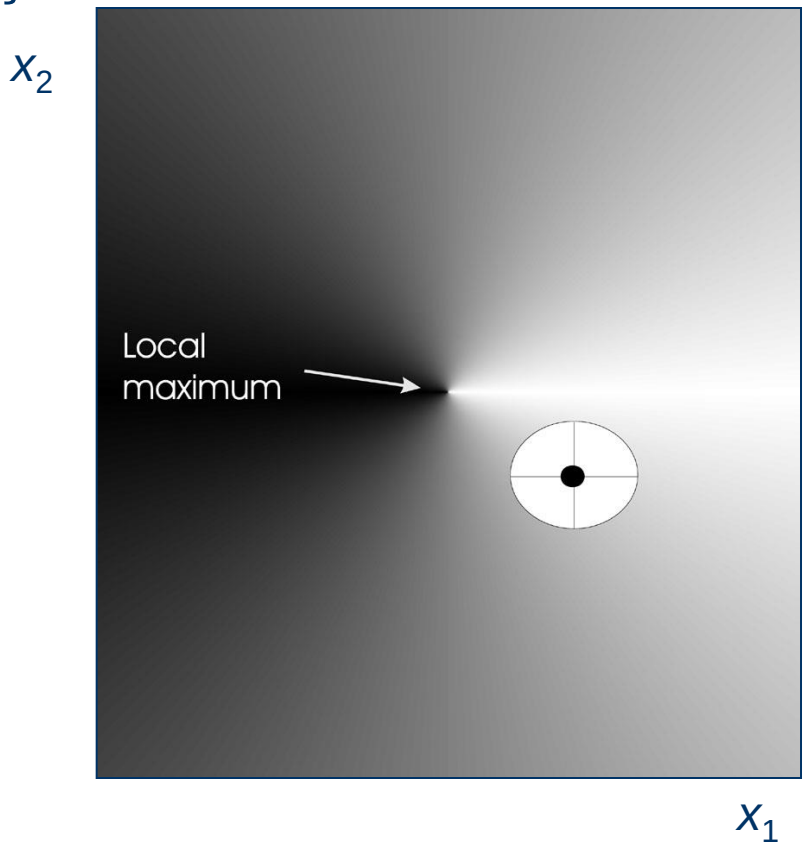
  - Parâmetro de mutação (*mutation step*): $\sigma$
    - Único para todos as dimensões do espaço de busca
    - Pode ser adaptável

# 1.3.3. Mutação



# 1.3.3. Mutação

- Mutação Gaussiana Não-correlacionada com um parâmetro de mutação
  - Exemplo



# 1.3.3. Mutação

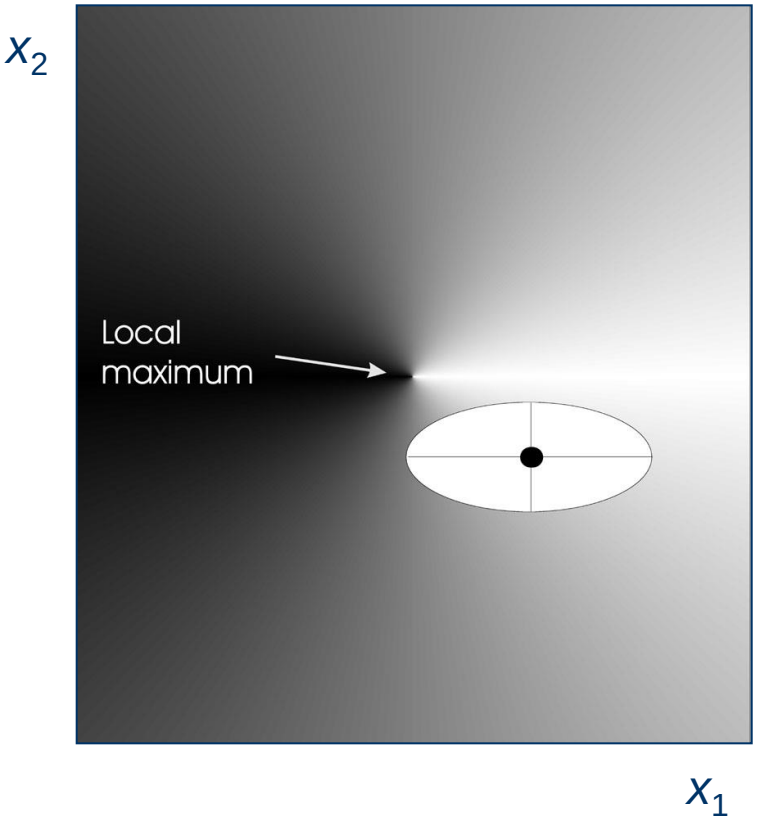
- Mutação Gaussiana Não-correlacionada com um vetor de parâmetros de mutação
  - Matriz **C** definida por

$$\mathbf{C} = \mathrm{diag}(\boldsymbol{\sigma})$$

  - Vetor de parâmetros de mutação: $\boldsymbol{\sigma}$
    - Um elemento do vetor para cada dimensão do espaço de busca
      - Definem diferentes passos de mutação para cada dimensão
    - Pode ser adaptável

# 1.3.3. Mutação

- Mutação Gaussiana Não-correlacionada com um vetor de parâmetros de mutação
  - Exemplo



# 1.3.3. Mutação

- Mutação Gaussiana Correlacionada
  - Matriz **C** definida como

    $c_{ii} = \sigma_i^2$

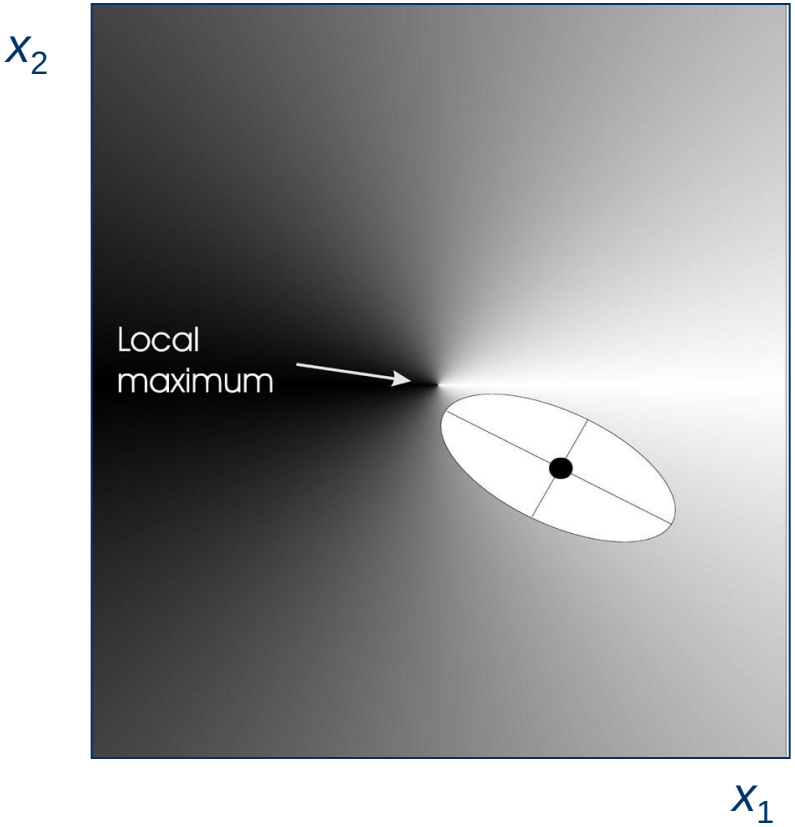
    $c_{ij} = 0$ se $i$ e $j$ não são correlacionados

    $c_{ij} = ½ \ (\sigma_i^2 - \sigma_j^2) \ \tan(2 \ \alpha_{ij})$ se $i$ e $j$ são correlacionados

    - Termo calculado através do uso de equações de rotação
  - Vetores de parâmetros de mutação: **σ** e **α**
    - O vetor **σ** tem tamanho $n$ (dimensão do espaço de busca)
    - O vetor **α** tem tamanho $k = n \ (n-1)/2$
    - Podem ser adaptáveis
  - Problema: muitos parâmetros a serem ajustados

# 1.3.3. Mutação

- Mutação Gaussiana Correlacionada
  - Exemplo

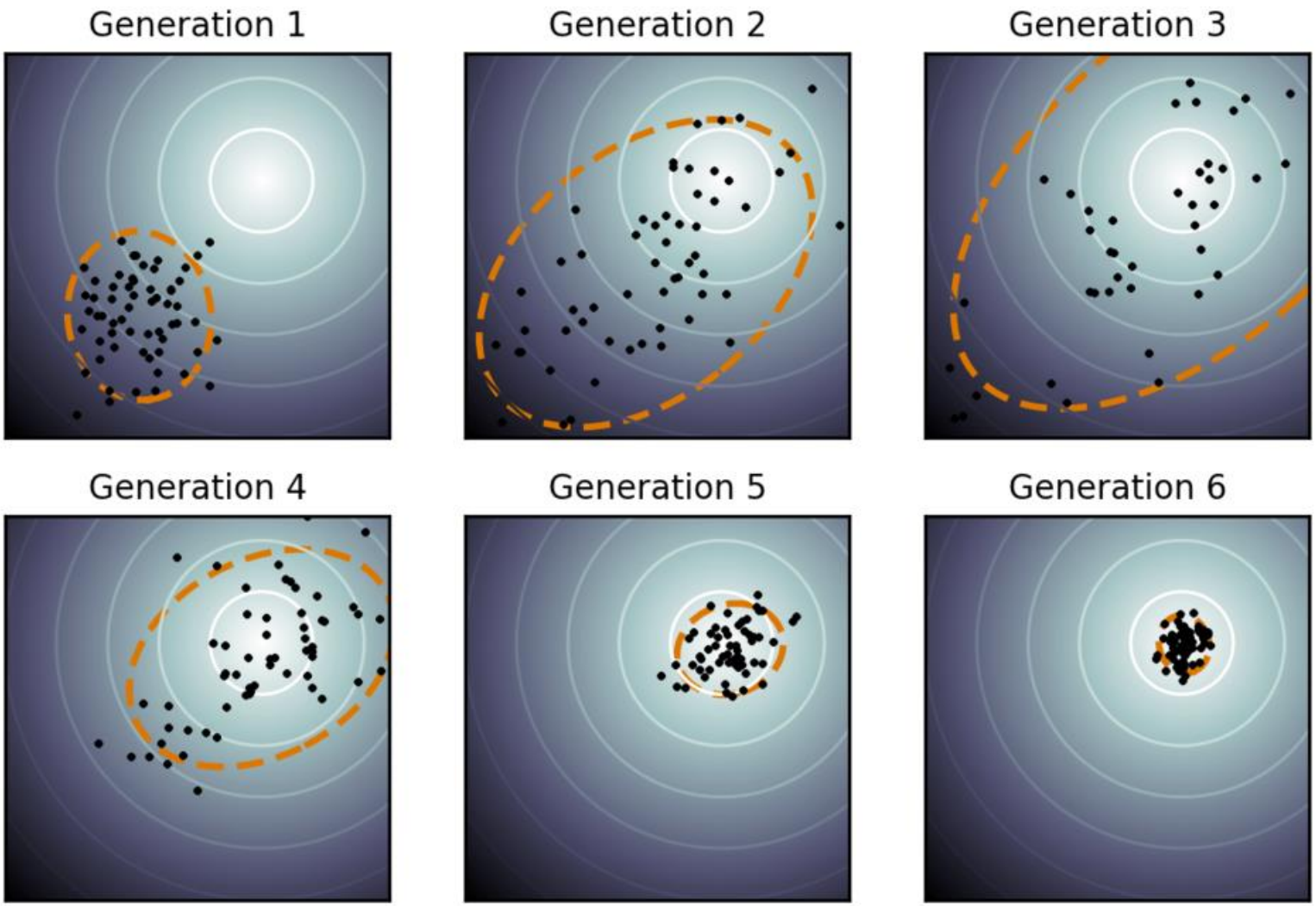


# 1.3.3. Mutação

- *Covariance Matrix Adaptation Evolution Strategy* (**CMA-ES**)
  - A média da distribuição é atualizada de modo que a probabilidade de soluções candidatas que previamente apresentaram sucesso é maximizada
  - Os elementos da matriz de covariância **C** são incrementadas de tal forma que a probabilidade de repetir mudanças nos passos de busca que apresentaram sucesso é aumentada
    - Para isso, a evolução da distribuição no tempo é armazenada (*evolution paths*)

# 1.3.3. Mutação

- CMA-ES
  - Assim, de maneira indireta, estima-se o gradiente da superfície de fitness
  - CMA-ES tem sido um dos algoritmos de maior sucesso em diversas aplicações de otimização contínua
    - Em particular, em problemas com funções de avaliação não-convexas, mal-condicionadas, multi-modais e/ou com ruído




Fonte: https://en.wikipedia.org/wiki/File:Concept_of_directional_optimization_in_CMA-ES_algorithm.png
Image based on work by Nikolaus Hansen and others.

# 1.3.4. Representação

- Cromossomos consistem de até três partes
  - Variáveis do objeto: $x_1,\ldots,x_n$
    - Vetor **x**
    - Sempre presente
  - 2 vetores de parâmetros de mutação (*strategy parameters*)
    - Tamanhos dos passos de mutação: $\sigma_1,\ldots,\sigma_n$
      - Pode usar menos do que *n* parâmetros
    - Ângulos de rotação: $\alpha_1,\ldots,\alpha_k$
      - Sendo $k = n\ (n-1)/2$
      - Pode usar menos do que *k* ângulos
- Vetor completo: $[\,x_1,\ldots,x_n, \sigma_1,\ldots,\sigma_n, \alpha_1,\ldots,\alpha_k\,]$

# 1.3.5. Recombinação

- Diferentemente dos AGs, gera apenas um filho de cada vez
- Dois tipos principais
  - Intermediária
  - Discreta

# 1.3.5. Recombinação

- Intermediária

$$x_g = \frac{1}{\rho}\sum_{i=1}^{\rho} x_g(i)$$

na qual

*g* indica o índice do elemento do vetor **x**

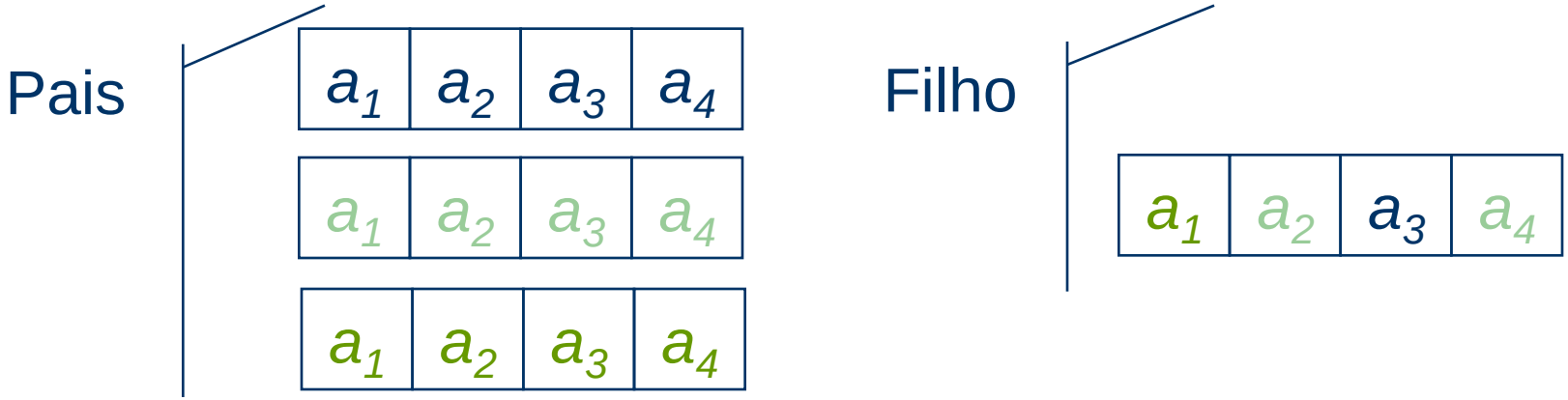
*i* indica o índice do pai

*ρ* indica o número de pais utilizado para gerar um filho

– Geralmente 2 ou *μ*

# 1.3.5. Recombinação

- Discreta (ou dominante)
  - Os genes são escolhidos aleatóriamente (com distribuição uniforme) entre os genes do $\rho$ pais
  - Exemplo:



# 1.3.6. Auto-Adaptação

- Parâmetros de mutação são permitidos evoluir também
  - Mutação Gaussiana Não-correlacionada com um parâmetro de mutação
    - O parâmetro que formence o passo de mutação $\sigma$ co-evolui junto com o vetor **x**

$$\sigma' = \sigma \exp(\tau \; N(0,1))$$

    - Tipicamente, a "taxa de aprendizado" é

$$\tau \propto 1/ n^{1/2}$$

    - Um limite inferior é estabelecido (válido para outros tipos também)

$$\sigma' < \varepsilon_0 \Rightarrow \sigma' = \varepsilon_0$$

    - Ordem é importante: primeiro $\sigma$ e depois **x**

# 1.3.6. Auto-Adaptação

- Parâmetros de mutação são permitidos evoluir também
  - Mutação Gaussiana Não-correlacionada com um vetor de parâmetros de mutação

$$\sigma'_i = \sigma_i \exp(\tau' N(0,1) + \tau N_i(0,1))$$

- Tipicamente, as “taxas de aprendizado” são

$$\tau' \propto 1/(2n)^{1/2}$$

$$\tau \propto 1/(2n^{1/2})^{1/2}$$

# 1.3.6. Auto-Adaptação

- Parâmetros de mutação são permitidos evoluir também
  - Mutação Gaussiana Correlacionada

$$\sigma'_i = \sigma_i \exp(\tau' N(0,1) + \tau N_i(0,1))$$

$$\alpha'_j = \alpha_j + \beta N(0,1)$$

  - Tipicamente, as “taxas de aprendizado” são

$$\tau' \propto 1/(2n)^{1/2}$$

$$\tau \propto 1/(2n^{1/2})^{1/2}$$

$$\beta \approx 5°$$

  - Limite inferior para vetor **α**

$$|\alpha'_j| > \pi \Rightarrow \alpha'_j = \alpha'_j - 2\pi \operatorname{sign}(\alpha'_j)$$

# 1.3.6. Auto-Adaptação

- Pré-requisitos
  - $\mu > 1$ : para testar diferentes estratégias
  - $\lambda > \mu$ : mais filhos do que pais
  - Pressão seletiva não demasiadamente alta, e.g., $\lambda \approx 7 \ \mu$
  - $(\mu,\lambda)$-ES : diminuir o problema de parâmetros de mutação mal-adaptados
  - Recombinação dos parâmetros de mutação através de recombinação (intermediária)

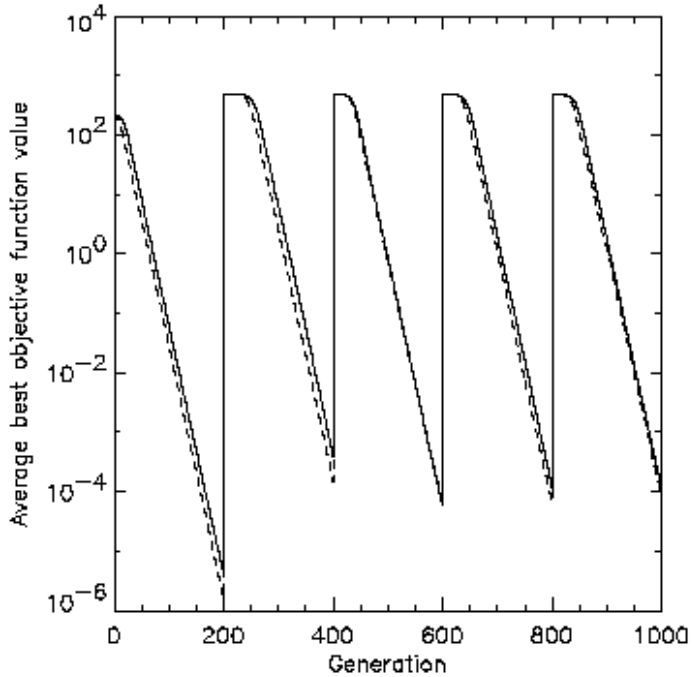
# 1.3.7. Exemplos

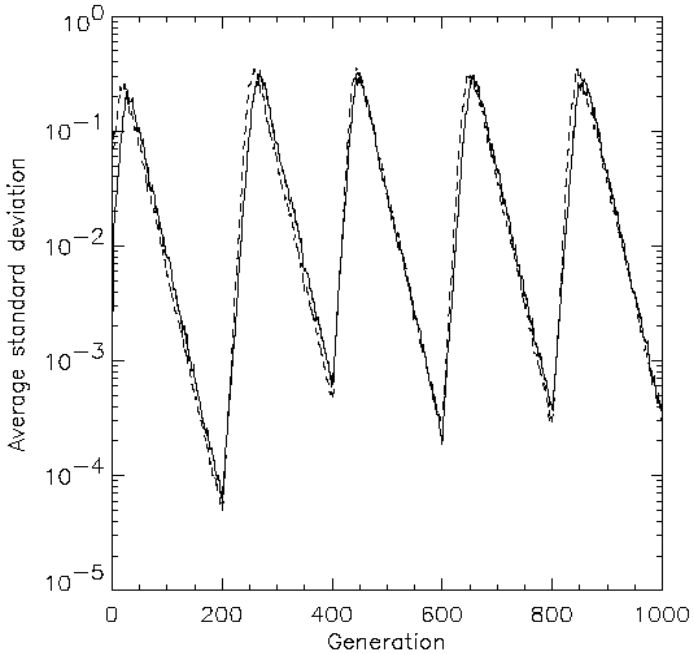
Exemplo 1.3.2.

- Problema de otimização dinâmica
  - Localização do ótimo é alterada a cada 200 gerações

# 1.3.7. Exemplos

## Exemplo 1.3.2.




Mudanças no fitness (esquerda) e nos parâmetros que definem os passos de mutação (direita)

# 1.3.7. Exemplos

Exemplo 1.3.3. (Eiben &Smith , 2003)

- Função Ackley com $n$ = 30

$$f(x) = -20 \cdot \exp\left( -0.2\sqrt{\frac{1}{n} \cdot \sum_{i=1}^{n} x_i^2} \right) - \exp\left( \frac{1}{n} \sum_{i=1}^{n} \cos(2\pi x_i) \right) + 20 + e$$

- Representação:
  - $-30 < x_i < 30$
  - 30 parâmetros de mutação (vetor **σ**)
- (30,200)-ES
- Critério de parada : depois de 200000 avaliações de fitness
- Resultados em 10 execuções:
  - Ótimo global foi encontrado em todas as execuções
  - Média do fitness da melhor solução final: $7{,}48 \quad 10^{-8}$

# 1.3.7. Exemplos

**2.2.3.** *$F_8$: Shifted Rotated Ackley's Function with Global Optimum on Bounds*

$$F_8(\mathbf{x}) = -20\exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} z_i^2}) - \exp(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi z_i)) + 20 + e + f_bias_8,\ \mathbf{z} = (\mathbf{x} - \mathbf{o}) * \mathbf{M},$$

$\mathbf{x} = [x_1, x_2, \ldots, x_D]$, $D$: dimensions

$\mathbf{o} = [o_1, o_2, \ldots, o_D]$ : the shifted global optimum;

After load the data file, set $o_{2j-1} = -32\ o_{2j}$ are randomly distributed in the search range, for $j = 1, 2, \ldots, \lfloor D/2 \rfloor$

**M**: linear transformation matrix, condition number=100

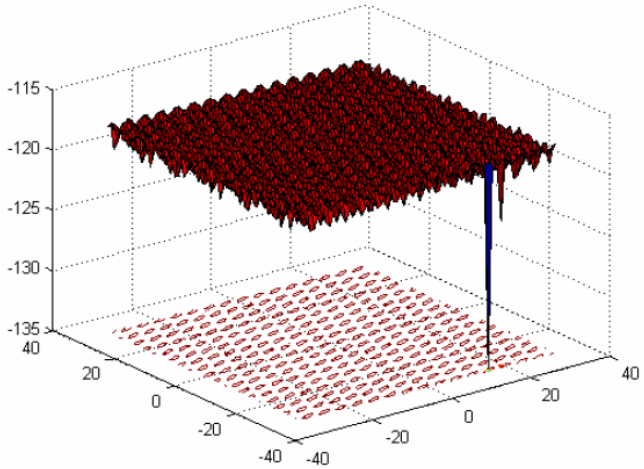



**Figure 2-8** 3-*D* map for 2-*D* function

P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y. P. Chen, A. Auger, and S. Tiwari. Problem definitions and evaluation criteria for the CEC 2005 Special Session on Real Parameter Optimization. *Technical Report*, Nanyang Technological University, 2005.

# Comentários

- Referências
  - Eiben, A. E. & Smith, J. E. *Introduction to Evolutionary Computation.* Springer, 2003
    - Capítulo 4
  - Beyer, H.-G. and Schwefel, H.-P. (2002). Evolution strategies: A comprehensive introduction. *Natural Computing,* 1: 3–52.
- Agradecimentos
  - Parte do material desta apresentação foi obtida através de
    - Material de apoio do livro Eiben, A. E. & Smith, J. E. *Introduction to Evolutionary Computation.* Springer, 2003.