



# Otimização de Dispositivos Eletromagnéticos pelo Método dos Elementos Finitos : os Métodos Estocásticos

Luiz Lebensztajn

# Organização

- Descrição de métodos de otimização estocásticos em um contexto de Cálculo de Campo por Elementos Finitos
- As vantagens e desvantagens da abordagem

# Métodos de Otimização: Características Típicas

## DETERMINÍSTICOS

- UTILIZAM DERIVADAS
- PONTO DE PARTIDA É IMPORTANTE
- NÚMERO REDUZIDO DE AVALIAÇÕES DA FUNÇÃO OBJETIVO
- ESTAGNAR-SE NUM MÍNIMO LOCAL

## ESTOCÁSTICOS

- NÃO UTILIZAM DERIVADAS
- EM GERAL TRABALHAM COM POPULAÇÕES
- NÚMERO ELEVADO DE AVALIAÇÕES DA FUNÇÃO OBJETIVO
- “GARANTIA” DO ÓTIMO GLOBAL

# Métodos de Otimização e Elementos Finitos

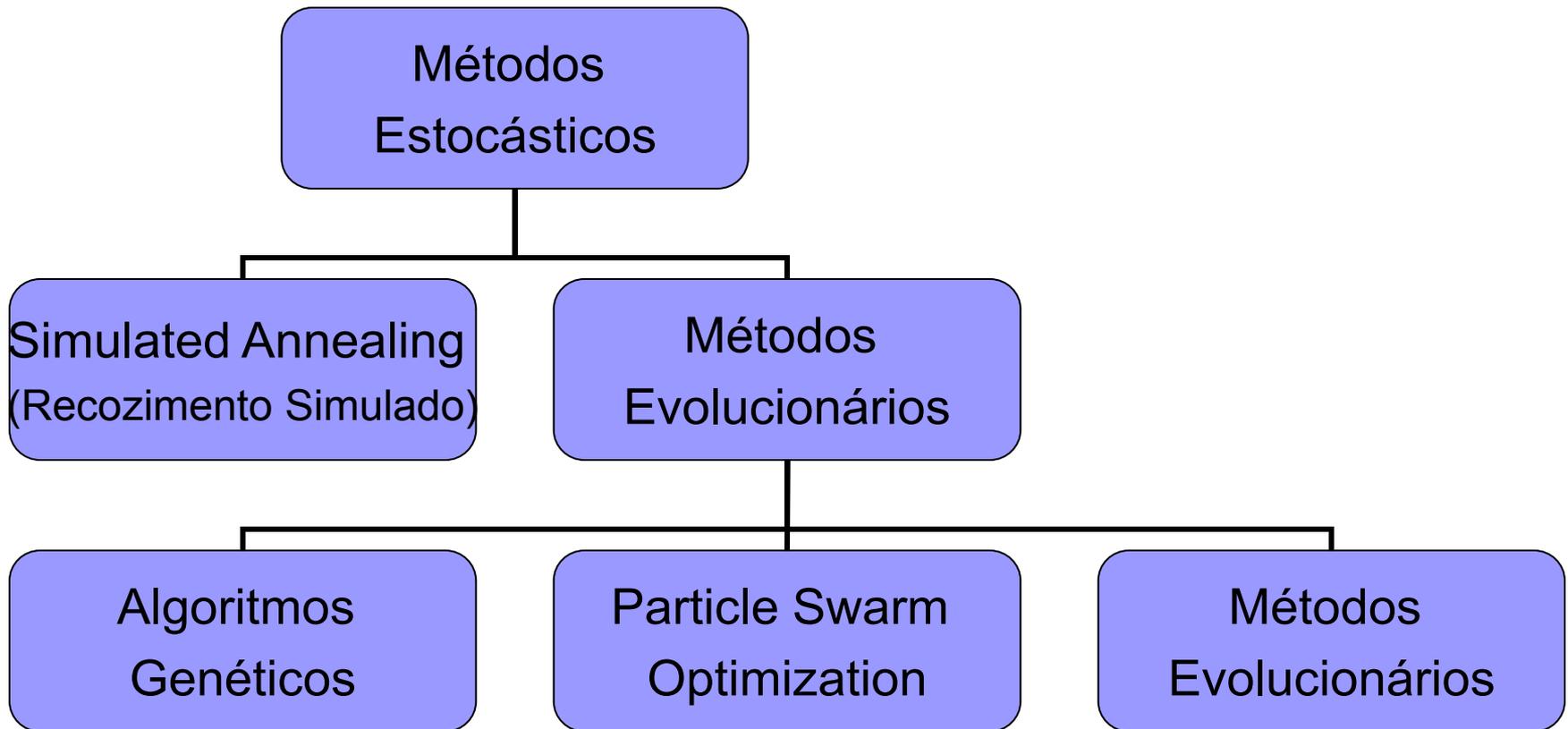
- Métodos determinísticos
  - Utilizam derivadas
  - Ponto de partida é importante
  - Número reduzido de avaliações da função objetivo
  - Pode estagnar-se em mínimo local

- Métodos estocásticos
  - Não utilizam derivadas
  - Em geral trabalham com populações
  - Número elevado de avaliações da função objetivo
  - “Garantia” de mínimo global

# Métodos de Otimização e Elementos Finitos

- Aparentemente os métodos estocásticos são mais atraentes para alcançar extremos de funções objetivo.
- Eles têm a desvantagem de um alto número de chamadas de função objetivo.
- Deve-se adotar algum procedimento para que o custo computacional não seja elevado.

# Alguns métodos estocásticos de otimização



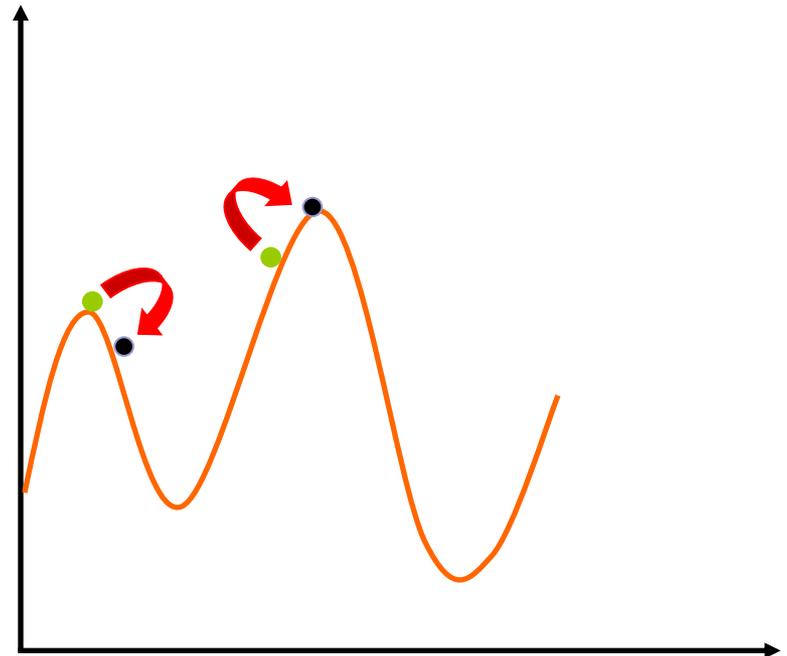
# Simulated Annealing

- PARTE DE UM ÚNICO INDIVÍDUO  
(GERADO ALEATORIAMENTE )

- ACEITA-SE UM MOVIMENTO SE:

$$\delta = f(E_f) - f(E_i) \leq 0$$

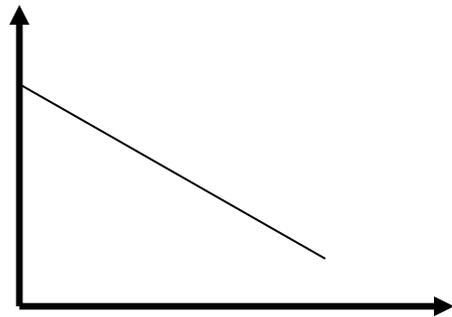
$$\text{random}(0,1) < \exp\left(-\frac{\delta}{T}\right)$$



- DIFICULDADE: ENCONTRAR A MELHOR MANEIRA DE  
REALIZAR O DECRESCIMENTO DA TEMPERATURA T

# Simulated Annealing

Temperatura



Iteração

- No início a probabilidade de aceite de um movimento do tipo up-hill deve ser alta → T alta
- No fim do processo, a probabilidade de aceite de um movimento up-hill deve ser pequena → T pequena

# Algoritmo Genético: um método estocástico típico

- Algoritmo Genético requer:
  - Uma representação genética do domínio de solução
  - Uma função de aptidão para avaliar o domínio de solução
- Um exemplo simples
  - minimizar  $y=(x-4)^2$  no intervalo  $0 < x < 63$
  - Uma possível representação (binária):
    - $x =$  
    - $x$  possui 6 posições e cada uma pode valer 0 ou 1.
    - transforma-se a representação binária de  $x$  em real
    - através de  $y$ , avalia-se a aptidão do elemento  $x$

# Minimizar $(x-4)^2$

- Uma população

1	0	1	1	1	0
0	0	1	1	1	0
1	0	0	0	1	0
1	0	1	1	1	1
0	0	1	0	0	1
1	1	1	1	1	0

# Minimizar $(x-4)^2$

## ■ População avaliada

1	0	1	1	1	0
0	0	1	1	1	0
1	0	0	0	1	0
1	0	1	1	1	1
0	0	1	0	0	1
1	1	1	1	1	0

x	y
46	1764
14	100
34	900
47	1849
9	25
62	3364

Seleção



# Minimizar $(x-4)^2$ -- Tipos de Seleção

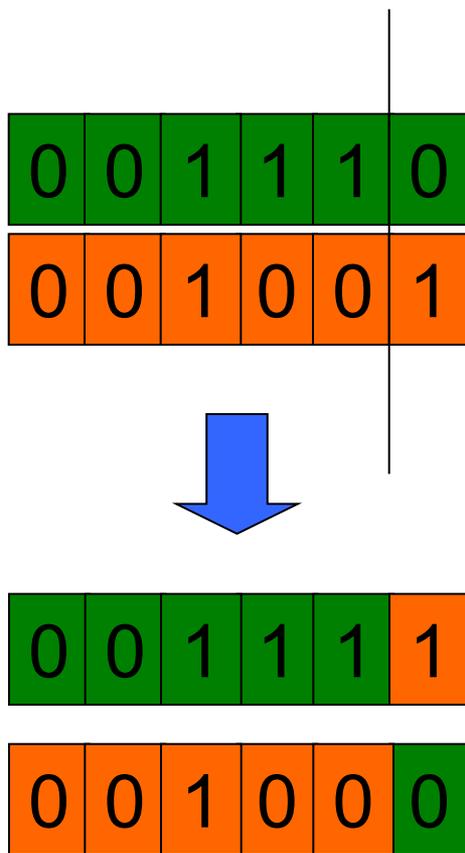
## ■ Roleta

- Proporcional ao valor da função
- Mais aptos são os mais prováveis para crossover.
- Convergência prematura >> Pressão da seleção

## ■ Torneio

- Faz-se k torneios de 2 indivíduos da população.
- O melhor dos 2 indivíduos é mantido para o conjunto de pais >> crossover.

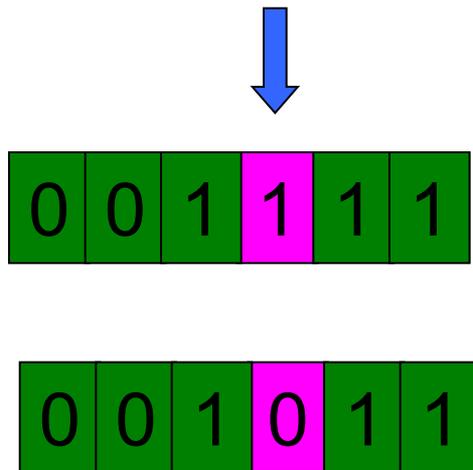
# Minimizar $(x-4)^2$ o Crossover



## Mecanismo do Crossover

- 001000 possui aptidão  $y=16$ , melhor que todos os indivíduos da população
- Note que mesmo 001111 é de alta qualidade (aptidão  $y=121$ )

# Minimizar $(x-4)^2$ – a mutação



## Mecanismo de Mutação

- Com uma baixa probabilidade cada bit é exposto ao processo de mutação.
- A aptidão passa de 121 para 49.
- Mutação → Inserção de nova informação genética

# Valores típicos em problemas reais

- Probabilidade de Mutação entre 0 a 5%
- Probabilidade de Crossover entre 70 a 100%
- Tamanho de população superior a 20
- Número de iterações entre 100 e 500

# Estratégia de Evolução

- Algoritmo de otimização que mescla dois outros algoritmos: o genético e o simulated annealing
- Ao mesmo tempo em que se busca imitar o desenvolvimento de populações, de geração em geração através de reprodução e mutação, à procura indivíduos cada vez melhores, faz-se um controle do quanto se muda de uma geração para outra com passos baseados num processo de recozimento de metais, evitando a dispersão dos melhores indivíduos.

# Estratégia de Evolução

- Característica: pais ( $\mu$ ) e os filhos ( $\lambda$ ).
- De acordo com a lógica usada na evolução entre parentes e filhos, dois tipos principais de estratégia podem ser adotadas:
  - a estratégia “mais” ( $\mu+\lambda$ ): em que a próxima geração é formada a partir dos  $\mu$  pais e  $\lambda$  filhos;
  - a estratégia vírgula. ( $\mu,\lambda$ ): uma vez gerados os filhos, escolhe-se, destes, a mesma quantidade de pais para formar a próxima geração (nenhum pai sobrevive).

# Estratégia de Evolução – Vantagens Desvantagens

- Estratégia vírgula tem maior probabilidade de convergência pelo fato de poder escapar de mínimos locais ao descartar os pais. Porém tem convergência mais lenta, já que a experiência dos pais é perdida dado o descarte.
- A estratégia mais apresenta a vantagem de um curso de convergência garantido rumo ao ótimo local ou global (conforme o desejado) por causa do elitismo entre pais e filhos, mantendo, desta forma, a trilha do melhor resultado já encontrado.

# Estratégia de Evolução – Detalhamento

- 1. Inicialização: Criam-se aleatoriamente os primeiros pais e calcula-se o valor da função objetivo, com auxílio do MEF.
- 2. Reprodução: Gera-se filhos a partir dos pais. Alteram-se os pais (via mutação), obtendo-se os filhos;
- 3. Seleção: seleção dos melhores indivíduos. Após calcular-se o valor da função objetivo (via MEF), escolhem-se os melhores indivíduos, ou seja, aqueles que se aproximam mais do valor ótimo da função objetivo.

# Estratégia de Evolução – Detalhamento

- 4. Recozimento: mudança do passo de mutação. Alterando-se o passo da mutação (o quanto de mutação se aplica) consegue-se acelerar a velocidade de convergência e garante-se o ótimo.
- 5. Finalização: critério utilizado para finalizar o processamento. Se tal critério não for satisfeito, volta-se ao passo 2. Por exemplo, pode-se considerar o número de iterações realizadas ou o valor da função objetivo.

# Particle Swarm Optimization (PSO)

- O PSO guarda algumas similaridades com o Algoritmo Genético.
- O sistema é inicializado com uma população de soluções aleatórias e busca o ótimo através de atualizações consecutivas.
- Mas o PSO não possui operadores de evolução como crossover ou mutação.
- Ele se baseia no comportamento social de enxames de abelhas ou cardumes de peixes.

# Particle Swarm Optimization (PSO)

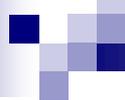
- No PSO, as soluções potenciais (denominadas partículas) fazem um “vôo” ao longo do espaço de definição das variáveis de otimização.
- Este vôo é dirigido pelas melhores partículas.

# Particle Swarm Optimization (PSO)

- Cada partícula possui:
  - seu valor de coordenadas no espaço das variáveis e seu valor atual de função objetivo.
  - armazena-se também a melhor função objetivo (e respectiva posição/ que esta partícula já atingiu). Este valor é denominado  $p_{best}$ .
  - O melhor local atingido por uma das partículas ( $l_{best}$ ) é também armazenado e possui valor  $g_{best}$ .

# Particle Swarm Optimization (PSO)

- O PSO consiste em uma seqüência de atualizações a cada instante de tempo. Muda-se a velocidade (de aceleração) de cada partícula a partir de pbest e lbest.
- $\mathbf{x}_{i+1} = \mathbf{x}_i + \mathbf{v}_i$
- $\mathbf{v}_{i+1} = \omega \mathbf{v}_i + c_1 r_{d1} (\mathbf{p}_{best} - \mathbf{x}_i) + c_2 r_{d2} (\mathbf{g}_{best} - \mathbf{x}_i)$
- $\omega$  - “constante de inércia”
- $c1$  e  $c2$  são constantes, valores típicos [1, 2]
- $r_{d1}$  e  $r_{d2}$  números aleatórios



# Particle Swarm

## Optimization >> Visualização

■ <http://www.projectcomputing.com/resources/psovis/index.html>

# ■ Evolução Diferencial

- É um método paralelo de busca direta pela solução global otimizada que utiliza uma população de vetores  $x_{i,G}$ , com dimensão  $D$  ( $D = n^\circ$  de parâmetros do problema).
  - NP= Tamanho da População
  - G = Geração/iteração da população
- O tamanho da população é fixo em cada geração.

- Existem as seguintes etapas de “evolução”:

- Seleção
- Mutação
- Crossover

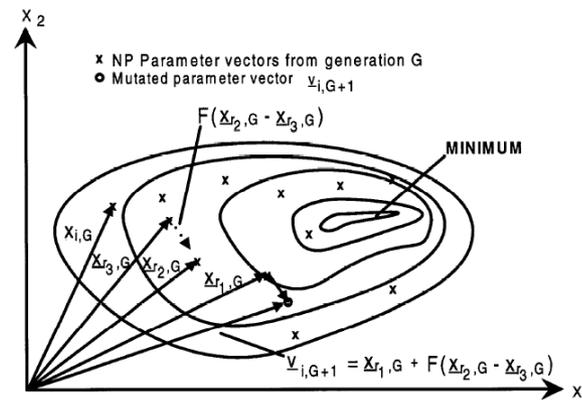


Figure 1. An example of a two-dimensional cost function showing its contour lines and the process for generating  $v_{i,G+1}$ .

- $F \Rightarrow$  Fator de Ponderação (número aleatório  $[0,1]$ )
- $CR \Rightarrow$  Probabilidade de Crossover

## ■ Mutaç o:

- Tr s vetores diferentes selecionados aleatoriamente da popula o formam um novo vetor  $v_{i,G}$ .

$$v_{i,G+1} = x_{r_1,G} + F \cdot (x_{r_2,G} - x_{r_3,G})$$

## ■ Crossover

- Um novo vetor  $u_{i,G}$    criado a partir da troca de par metros entre os vetores  $x_{i,G}$  e  $v_{i,G}$ .

$$u_{ji,G+1} = \begin{cases} v_{ji,G+1} & \text{if } (randb(j) \leq CR) \text{ or } j = rnbr(i) \\ x_{ji,G} & \text{if } (randb(j) > CR) \text{ and } j \neq rnbr(i) \end{cases}$$

$$j = 1, 2, \dots, D.$$

## ■ Sele o:

- $X_{i,G+1} = \min(\text{fun\_obj}(x_{i,G}); \text{fun\_obj}(u_{i,G}))$

■ O DE possui variantes identificadas por DE x/y/z

- x -> Vetor de mutação (“rand” ou “best”)
- y -> Número de vetores na diferença
- z -> Tipo de crossover (“bin” ou “exp”)

## ■ Esquema DE/rand/1

$$\underline{v}_{i,G+1} = \underline{x}_{r_1,G} + F \cdot (\underline{x}_{r_2,G} - \underline{x}_{r_3,G})$$

## ■ Esquema DE/best/1

$$\underline{v}_{i,G+1} = \underline{x}_{best,G} + F \cdot (\underline{x}_{r_1,G} - \underline{x}_{r_2,G}).$$

## ■ Esquema DE/best/2

$$\underline{v}_{i,G+1} = \underline{x}_{best,G} + F \cdot (\underline{x}_{r_1,G} + \underline{x}_{r_2,G} - \underline{x}_{r_3,G} - \underline{x}_{r_4,G}).$$

## ■ Esquema DE/rand to best/2

$$\underline{v}_{i,G+1} = \underline{x}_{i,G} + \lambda \cdot (\underline{x}_{best,G} - \underline{x}_{i,G}) + F \cdot (\underline{x}_{r_2,G} - \underline{x}_{r_3,G}).$$