



SISTEMAS INTELIGENTES

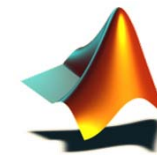
Prática 10 – Algoritmos Genéticos (Otimização de Sistemas)

Ivan Nunes da Silva



Objetivos da Aula

- Fixar a teoria sobre os processos de otimização envolvendo Algoritmos Genéticos, conforme visto nas aulas teóricas.
- Introduzir os principais componentes da ToolBox de Algoritmos Genéticos implementados no Matlab.
- Implementar soluções inteligentes para problemas envolvendo Otimização de Processos.



Definindo Problemas de Otimização

- Um problema geral de otimização pode ser definido pelas seguintes expressões:

Minimizar: $f(\mathbf{x})$ **{Função objetivo ou *fitness function*}**

sujeito a: $\mathbf{A} \cdot \mathbf{x} \leq \mathbf{b}$ **{Restrições de desigualdade}**

$\mathbf{A}^{\text{eq}} \cdot \mathbf{x} = \mathbf{b}^{\text{eq}}$ **{Restrições de igualdade}**

$\mathbf{x}^{\text{min}} \leq \mathbf{x} \leq \mathbf{x}^{\text{max}}$ **{Restrições de valores p/ *x* // *bounds constraints*}**

Onde: \mathbf{x} define o vetor das N variáveis do problema $\{\mathbf{x} \in \mathbb{R}^N\}$;

$f(\mathbf{x})$ define a função a ser otimizada;

\mathbf{A} define a matriz dos coeficientes das expressões de desigualdade, sendo \mathbf{b} o vetor de termos independentes;

\mathbf{A}^{eq} define a matriz dos coeficientes das expressões de igualdade, sendo \mathbf{b}^{eq} o vetor de termos independentes;

\mathbf{x}^{min} e \mathbf{x}^{max} são vetores que definem os valores mínimo e máximo para cada componente do vetor \mathbf{x} .

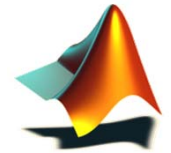
Exemplo {

$$\begin{cases} \text{Minimizar: } f(\mathbf{x}) = \cos(x_1) + 3x_2 - x_3 \\ \text{sujeito a: } x_1 + x_2 \leq 3 \\ \quad \quad \quad x_2 + 2x_3 \leq 0 \\ \quad \quad \quad x_1, x_2, x_3 \geq 0 \end{cases}$$

Parâmetros { $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$ $\mathbf{A} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 2 \end{bmatrix}$ $\mathbf{b} = \begin{bmatrix} 3 \\ 0 \end{bmatrix}$ $\mathbf{x}^{\text{min}} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$

3

Acessando a Toolbox de Algoritmos Genéticos



- A ToolBox de Algoritmos Genéticos (AG) do Matlab possui uma interface gráfica amigável e eficiente.
- De maneira similar aos outros ToolBoxes, a ToolBox de AG permite uma prototipagem extremamente rápida do problema a ser implementado por meio dos algoritmos genéticos.
- No Matlab, a otimização sempre é realizada com o intuito de minimizar a função objetivo. Caso haja necessidade de maximizar, basta-se lembrar que: $\text{Min } f(\mathbf{x}) = \text{Max } -f(\mathbf{x})$
- Para acessar a ToolBox de Algoritmos Genéticos no Matlab (versão 2011b ou superior), basta-se digitar “**optimtool('ga')**” na linha de comando.



Menu de seleção do otimizador

Janela de definição do problema

Botão para executar o AG

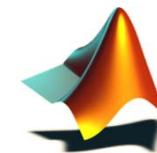
Janela de visualização de resultados

Janela de visualização da solução final

Janela de configuração do AG

Botão p/ expandir ou contrair o Help

The screenshot shows the MATLAB Optimization Tool window. The 'Solvers' dropdown is set to 'ga - Genetic Algorithm'. The 'Problem Setup and Results' section includes fields for 'Fitness function', 'Number of variables', and 'Constraints' (Linear inequalities, Linear equalities, Bounds, Nonlinear constraint function). The 'Options' section is expanded, showing 'Population' (type: Double Vector, size: 20), 'Fitness scaling' (Rank), 'Selection' (Stochastic uniform), 'Reproduction' (Elite count: 2, Crossover fraction: 0.8), and 'Mutation' (Use constraint dependent default). The 'Run solver and view results' section has 'Start', 'Pause', and 'Stop' buttons. The 'Quick Reference' panel on the right lists sections like 'Genetic Algorithm Solver', 'Problem Setup and Results', and 'Options' with expandable links.

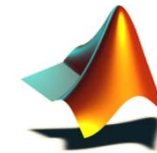


Problema 1:

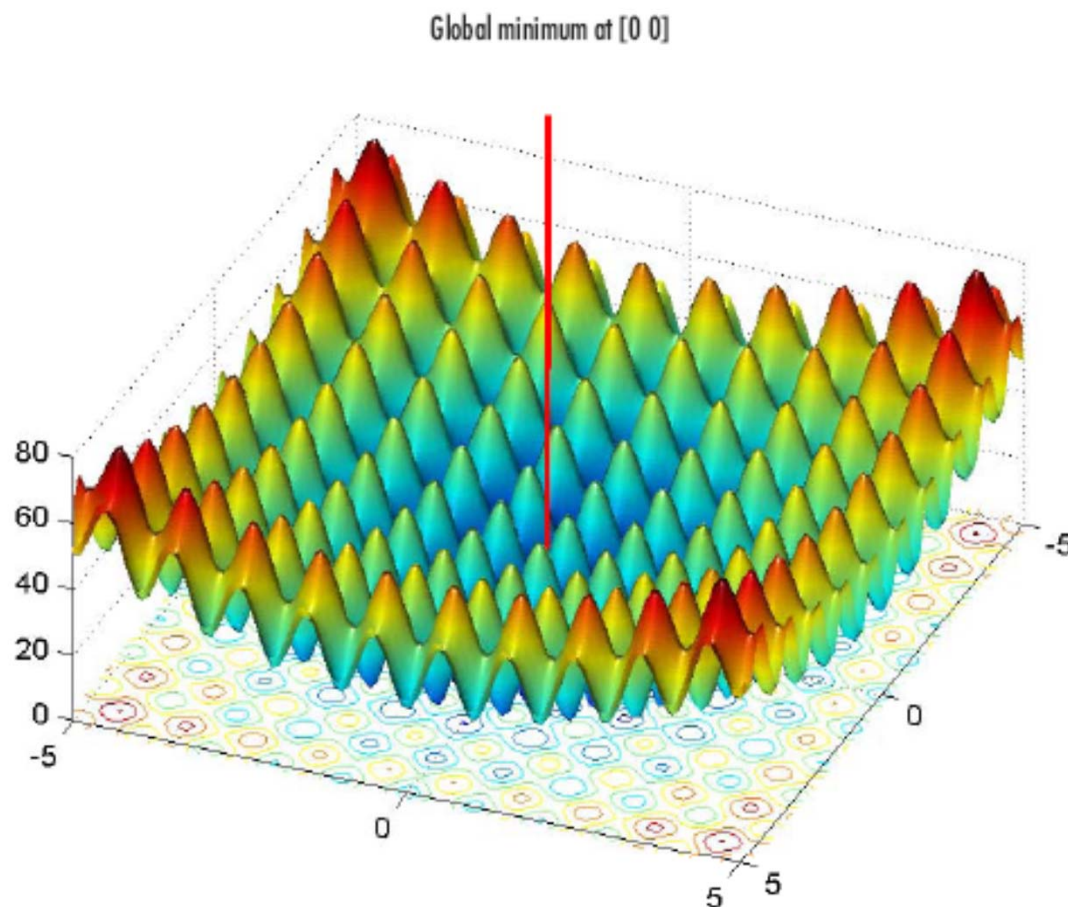
- **Seja o seguinte problema de otimização irrestrita, isto é, sem restrições, definido por:**

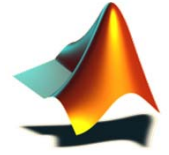
$$f(x_1, x_2) = 20 + x_1^2 + x_2^2 - 10 \cdot (\cos(2\pi x_1) + \cos(2\pi x_2))$$

- **Minimize a função acima por meio de um AG, com os seguintes parâmetros de configuração:**
 - **Tipo de indivíduos:** Real (double vector).
 - **Quantidade de indivíduos:** 100 (population size).
 - **Função de criação dos indivíduos:** uniforme no domínio (constraint dependent).
 - **Método de escalamento:** melhores indivíduos (rank).
 - **Método de Seleção:** amostragem estocástica.
 - **Método de Cruzamento (Reprodução):**
 - Taxa de elitismo igual a 2 (Elite count = 2) → É quantidade de indivíduos (de menor fitness) a passar diretamente p/ próxima geração;
 - Taxa de cruzamento de 85% (Crossover Fraction = 0.85).
 - **Método mutação:** em gene aleatório (constraint dependent).



- A função a ser otimizada é extremamente não-linear, sendo de difícil resolução por métodos convencionais, contendo diversos pontos de mínimos locais.
 - Mínimo Global → $\mathbf{x} = [0.0, 0.0]$





1º Passo: Criando a função de Fitness

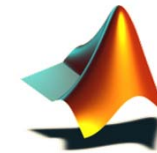
- A função de fitness é definida por:

$$f(x_1, x_2) = 20 + x_1^2 + x_2^2 - 10 \cdot (\cos(2\pi x_1) + \cos(2\pi x_2))$$

- Nesse caso, a mesma deve ser formulada a partir da especificação de sua respectiva função (“function” do matlab);
- Assim, implemente a função “f1genetico”, a qual receberá o vetor \mathbf{x} , constituído de dois argumentos [x1, x2], a fim de produzir a saída desejada y, isto é:

```
function y = f1genetico(x)
(...)
```

- Teste a função na linha de comando a fim de verificar se a mesma foi implementada corretamente:
 - Para $\mathbf{x} = [1, 1]$, a resposta deve ser 2.

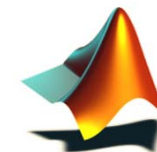


2º Passo: Inserindo a função de Fitness no Toolbox

- A inserção da função de fitness dentro da Toolbox é realizada pela associação de seu **ponteiro**, o qual é definido pelo operador “@”, isto é:

Problem	
Fitness function:	@f1genetico
Number of variables:	2

$$f1genetico(\mathbf{x}) = 20 + x_1^2 + x_2^2 - 10 \cdot (\cos(2\pi x_1) + \cos(2\pi x_2))$$



3º Passo: Inserindo os parâmetros de configuração do AG, conforme o slide 6, isto é:

Population

- Population type: **Double Vector (Domínio Real)**
- Population size: **100**
- Creation function: **Constraint Dependent:**
 - Uniforme → se não existe restrições
 - Feasible Population → cria população inicial respeitando os limites das variáveis

Fitness scaling

- Scaling function: **Rank (Ordenação baseada na aptidão do melhores indivíduos)**

Selection

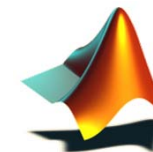
- Selection function: **Stochastic Uniform (Amostragem Uniforme Estocástica)**

Reproduction

- Elite count: **2 (Taxa de Elitismo → Os 2 melhores indivíduos serão também selecionados p/ a próxima geração)**
- Crossover fraction: **0.85 (Cruzamento em 85% da População)**

Mutation

- Mutation function: **Constraint dependent (otimiza a aleatoriedade do processo em função das restrições impostas).**



4º Passo: Definindo os parâmetros associados ao critério de parada do AG (assuma os valores defaults), isto é:

Stopping criteria

Generations: Use default: 100 Specify:

Time limit: Use default: Inf Specify:

Fitness limit: Use default: -Inf Specify:

Stall generations: Use default: 50 Specify:

Stall time limit: Use default: Inf Specify:

Function tolerance: Use default: 1e-6 Specify:

Nonlinear constraint tolerance: Use default: 1e-6 Specify:

Define uma quantidade máxima de gerações (iterações) como critério de parada do AG (Especificar: 150)

Define um tempo máximo de execução como critério de parada do AG (Usar default: Infinito)

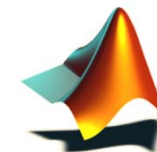
Define um valor mínimo de função objetivo que causará a parada do AG (Usar default: -Infinito)

Define uma quantidade máxima de seguidas iterações/gerações pobres (que não melhoram a função objetivo) como critério de parada

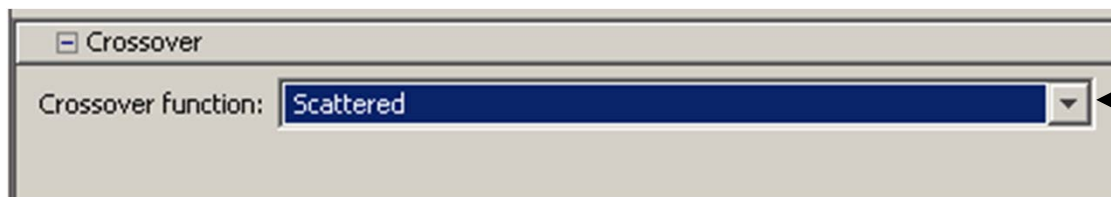
Define um limite tempo para que o AG consiga melhorar o valor da função objetivo.

Define uma tolerância a ser observada nos valores da função objetivo (entre sucessivas gerações) a fim de parar o AG. (Especificar: 10^{-8})

Define uma tolerância a ser observada nos valores das restrições do problema (entre sucessivas gerações) a fim de parar o AG.



5º Passo: Definindo a função de Cruzamento do AG



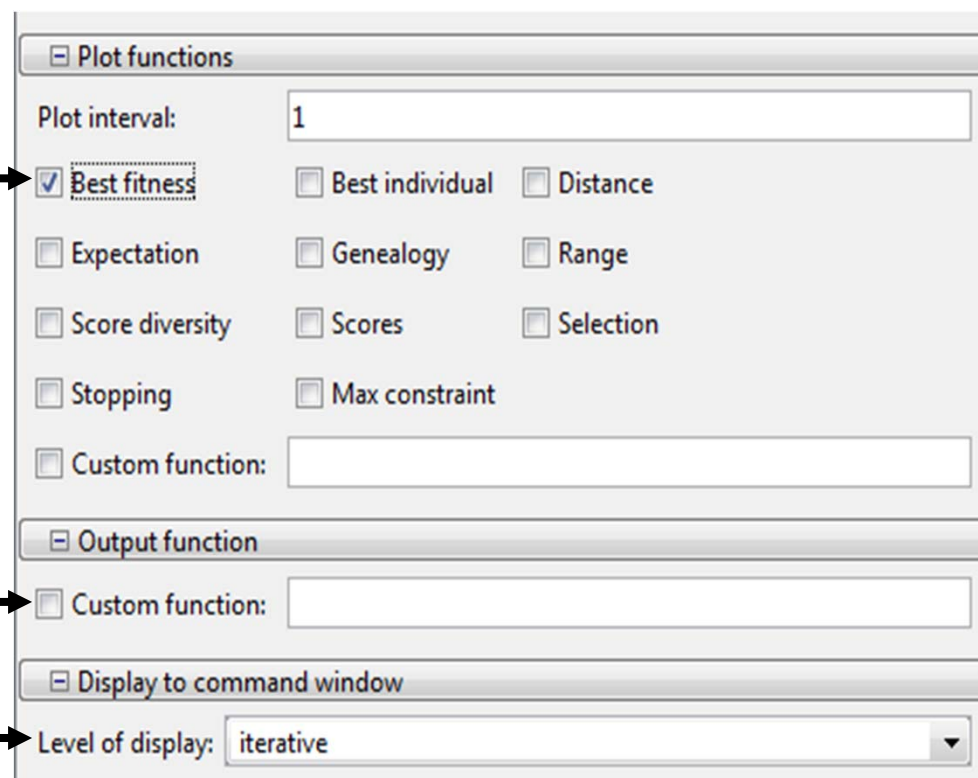
Scattered
(Método da Máscara)

6º Passo: Definindo formatos de saída do AG

Plotar o gráfico do melhor fitness em cada iteração (selecionar “Best fitness”)

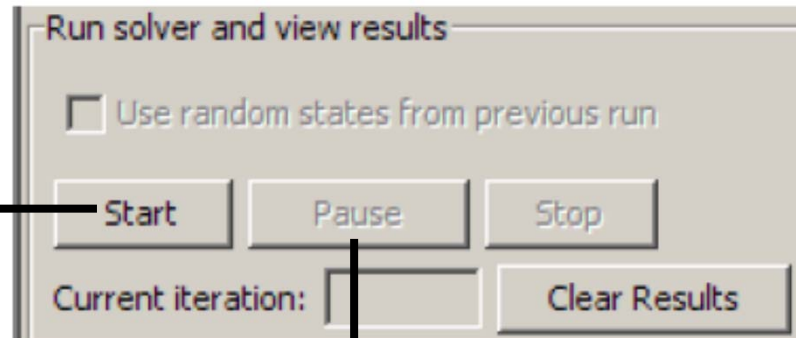
Fornecer os resultados dos valores da função fitness original (deixar desmarcado)

Mostrar evolução do AG na linha de comando (selecione “iterative”)





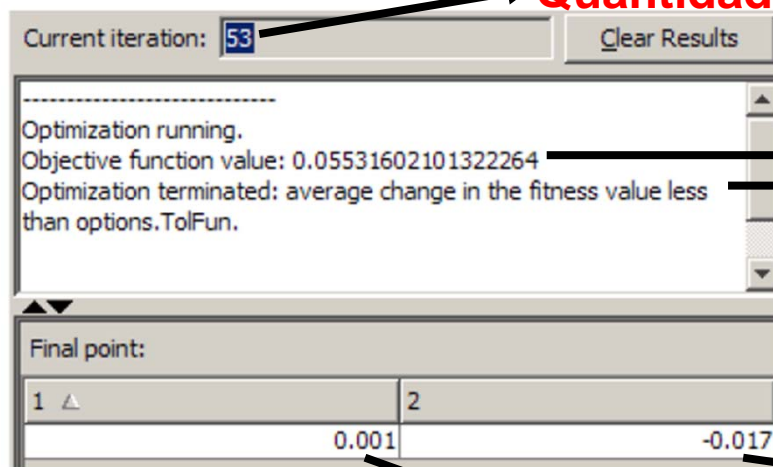
7º Passo: Executando o AG



Botão para executar o AG

Botão que realiza uma pausa no processamento do AG

8º Passo: Interpretando os Resultados



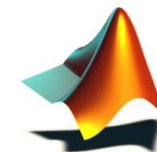
Quantidade de gerações atingidas

Valor final da função objetivo

Indicação do critério que causou a parada do AG

Valor final de x1

Valor final de x2



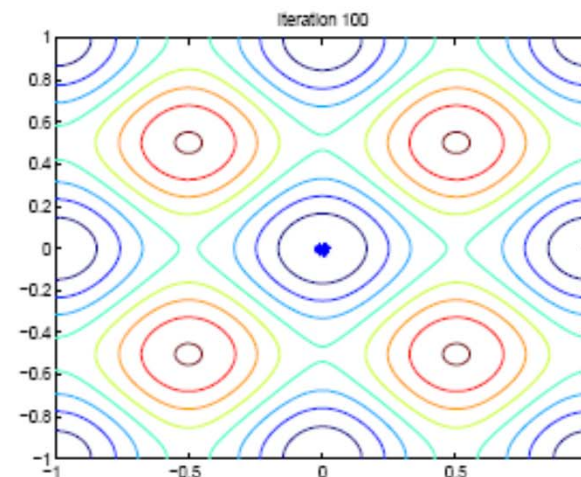
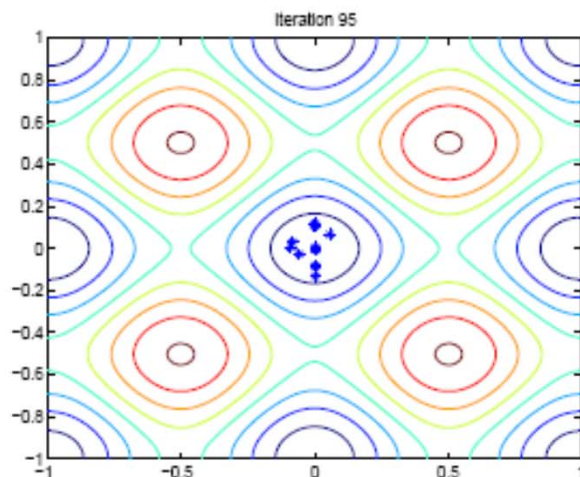
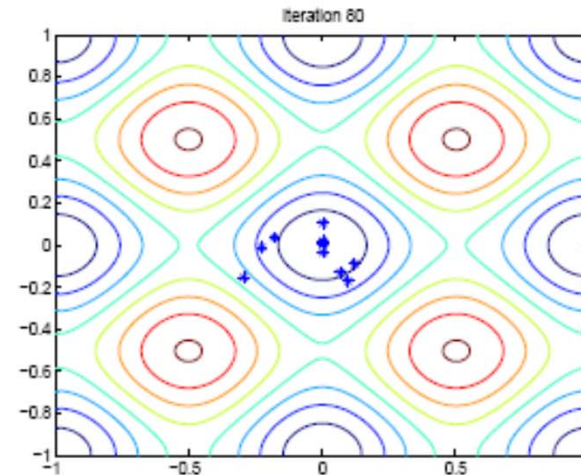
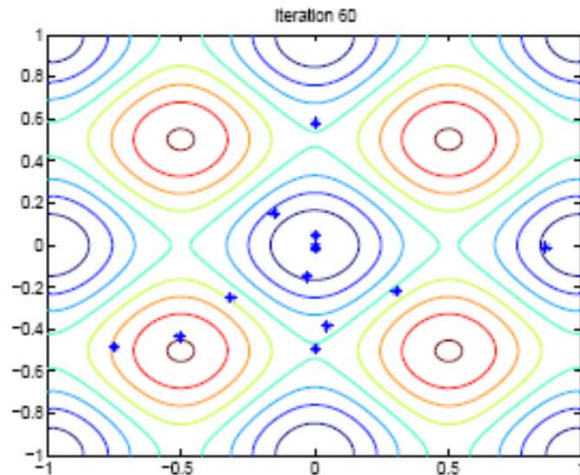
a) Realizar 5 execuções do AG para este problema, inserindo as respectivas respostas na tabela abaixo.

Execução	Número de Gerações	Valor de x_1	Valor de x_2	Valor da função objetivo
1				
2				
3				
4				
5				

b) Para a melhor resposta do item (a), trace o gráfico do valor da função objetivo em função da respectiva geração.



Ilustração do processo de convergência (Problema 1)



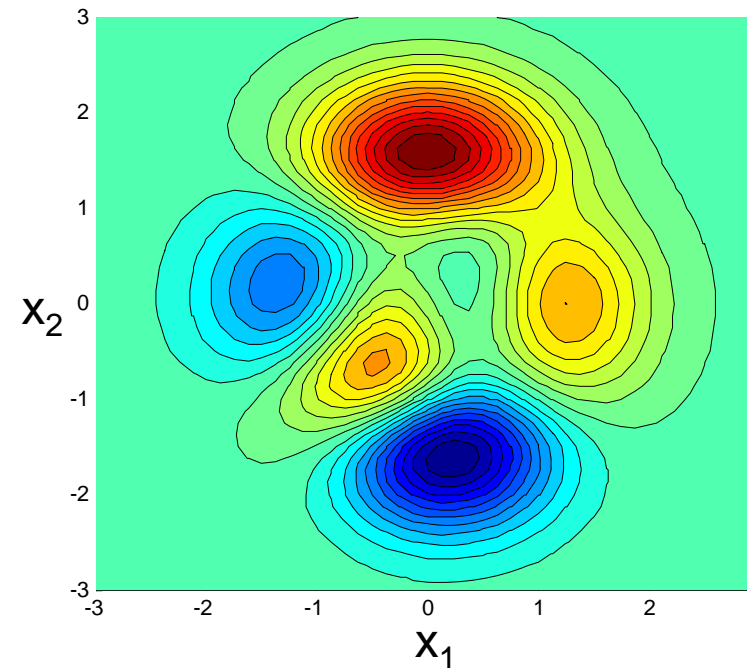
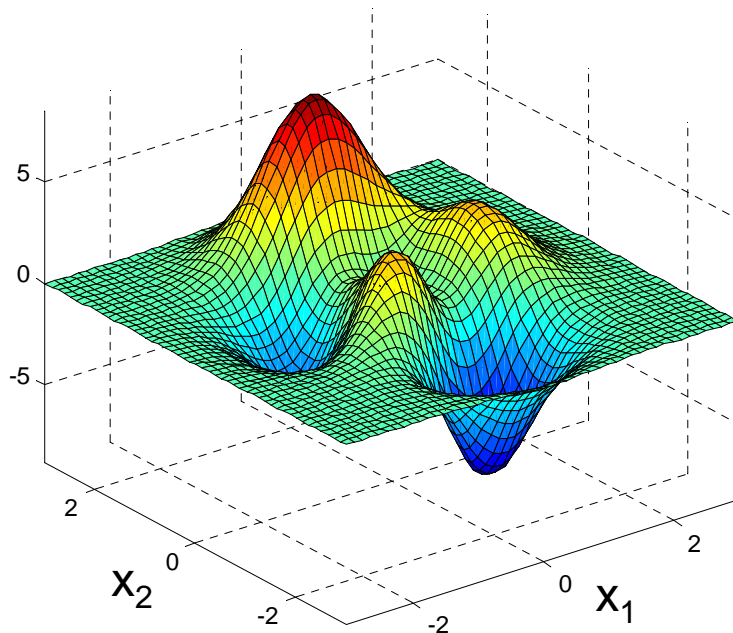


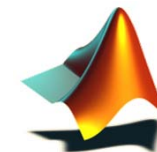
Problema 2:

➤ **Seja o seguinte problema de otimização:**

$$f(x_1, x_2) = 3 \cdot (1 - x_1)^2 \cdot e^{(-x_1^2 - (x_2 + 1)^2)} \dots$$

$$\dots - 10 \cdot \left(\frac{x_1}{5} - x_1^3 - x_2^5 \right) \cdot e^{(-x_1^2 - x_2^2)} - \frac{1}{3} e^{-(x_1 + 1)^2 - x_2^2}$$





- a) Obter o ponto de **máximo** da função “f2genetico” usando os mesmos parâmetros de configuração utilizado para o Problema 1, registrando as respectivas respostas na tabela abaixo.

Execução	Número de Gerações	Valor de x1	Valor de x2	Valor da função objetivo
1				
2				
3				
4				
5				

- b) Para a melhor resposta do item (a), trace o gráfico do valor da função objetivo em função da respectiva geração.

- c) Obter o ponto de **mínimo** da função “f2genetico”, com $-3 \leq x1 \leq -1$ e com $-1 \leq x2 \leq 1.5$.

Execução	Número de Gerações	Valor de x1	Valor de x2	Valor da função objetivo
1				
2				
3				
4				
5				



Ilustração do processo de convergência (Problema 2)

