

COMPUTAÇÃO EVOLUCIONÁRIA: INTRODUÇÃO

Prof. Cristiano Leite de Castro
crislcastro@ufmg.br

Departamento de Engenharia Elétrica
Universidade Federal de Minas Gerais
Belo Horizonte, Brasil

Dia do Mês	Aula	Assuntos
3/8/2016	1	Apresentação do Curso. Motivação e Introdução à Computação Evolucionária.
3/15/2016	1	Conceitos Básicos sobre algoritmos evolutivos (AE): Inspiração Biológica; Exemplo de Sistemas Evolutivo; Algoritmos Evolutivos.
3/22/2016	1	Lab1: 22/03/2016 (Cristiano + Tamires)
3/29/2016	0	Não haverá aula conforme calendário UFMG
4/5/2016	1	Introdução aos Algoritmos Genéticos (AG): Representação: Binário, Real. Operadores de Variação.
4/12/2016	1	Algoritmos Genéticos (AG): Operações de Seleção e Modelos de População. Elitismo.
4/19/2016	1	Lab2: 19/04/2016 (Tamires)
4/26/2016	1	Lab2: 26/04/2016 (Tamires)
5/3/2016	1	Estratégias Evolutivas (ES): Conceitos básicos. Distribuição normal uni e multivariada; Matriz de Covariância; Tipos de ES;
5/10/2016	1	Algoritmo de Evolução Diferencial (DE). Conceitos básicos. (Trabalho Prático Disponível)
5/17/2016	1	Lab 3: 10/11/2015 - (Tamires)
5/24/2016	1	Lab 3: 17/11/2015 (Tamires)
5/31/2016	1	Prova: 31/05/2016
6/7/2016	1	Dúvidas e Esclarecimentos sobre Trabalho Prático
6/14/2016	1	Lab 4: Entrega do Trabalho Prático (Tamires e Cristiano)
6/21/2016	1	Lab 4: Entrega do Trabalho Prático (Tamires e Cristiano)
6/28/2016	0	Exame Especial
Total de aulas	15	
		Aulas de Laboratório a serem realizadas na sala 310 do CCE (reservada)

- **Avaliações:**

- Provas (1 x 25%) = 25
- Trabalhos Práticos (1 x 30%) = 30
- Laboratórios (3 x 15%) = 45
- Total = 100

COMO ESTUDAR?

[1]	A.E. Eiben, J.E. Smith, Introduction to Evolutionary Computing (Natural Computing Series), Springer, 2010.
[2]	A. Gaspar-Cunha, R. Takahashi, C. Antunes. Manual de Computação Evolutiva e Metaheurística, Imprensa da Universidade de Coimbra/ Editora UFMG, 2012.
[3]	T. Back. Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming and genetic algorithms. New York: Oxford University Press, 1996.
[4]	C. Blum and A. Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. ACM Comput. Surv.35, 3 (September 2003), 268-308.
[5]	L. N. de Castro & J. Timmis, Artificial Immune Systems: A New Computational Intelligence Approach, Berlin, Germany: Springer-Verlag, 2002.
[6]	Notas de Aula do curso e Slides. (MOODLE)
[7]	KENNEDY, James F.; EBERHART, Russell C.; SHI, Yuhui. Swarm intelligence. San Francisco: Morgan Kaufmann Publishers, c2001. xxvii, 512 p. (The Morgan Kaufmann series in evolutionary computation) ISBN 1558605959.
[8]	FOGEL, David B. Evolutionary computation : toward a new philosophy of machine intelligence. New York: IEEE Press, c1995.

MONITORIA

- PROFESSORA ASSISTENTE:
 - Tamires Martins Rezende (Mestranda em E.E.)
 - Email: tamiresrezende@ufmg.br

PEQUENA REVISÃO SOBRE OTIMIZAÇÃO

- Como modelar um problema de otimização?
- Formalmente, um problema de otimização pode ser definido por:

$$\mathbf{x}^* = \min_{\mathbf{x}} f(\mathbf{x}) \text{ (ou max)} \quad f(\cdot): \mathbb{R}^n \rightarrow \mathbb{R}$$

$$\text{sujeito a: } \begin{cases} g_i(\mathbf{x}) \leq 0; & i = 1, \dots, p \\ h_j(\mathbf{x}) = 0; & j = 1, \dots, q \end{cases}$$

- Qual o significado da formulação anterior?

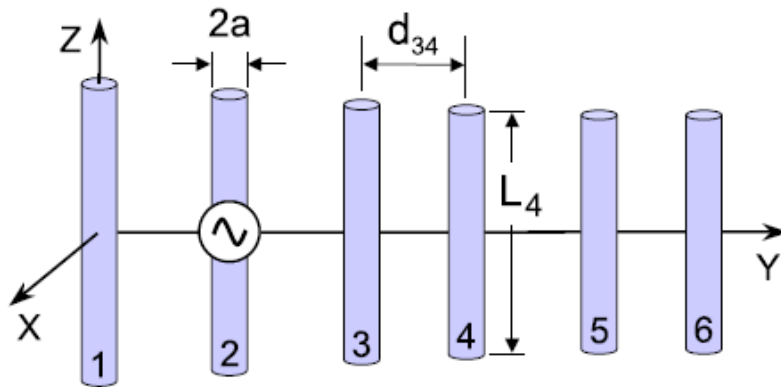
Obs.: variáveis em negrito são vetores; as demais são escalares.

- Ex.: projeto de uma antena:
 - **objetivo:** maximizar o desempenho de transmissão da antena, atendendo a um valor máximo requerido para o seu tamanho (comprimento);
 - matematicamente:

$$\begin{cases} \max: f(\mathbf{x}) = \text{desempenho}^* \\ \text{sujeito a: } g_1(\mathbf{x}): T \leq T_{max} \end{cases}$$

*o desempenho pode ser medido pela relação frente-costa da antena, geralmente medida em decibéis [Cunha et al, 2012];

- modelo 2D de uma antena:

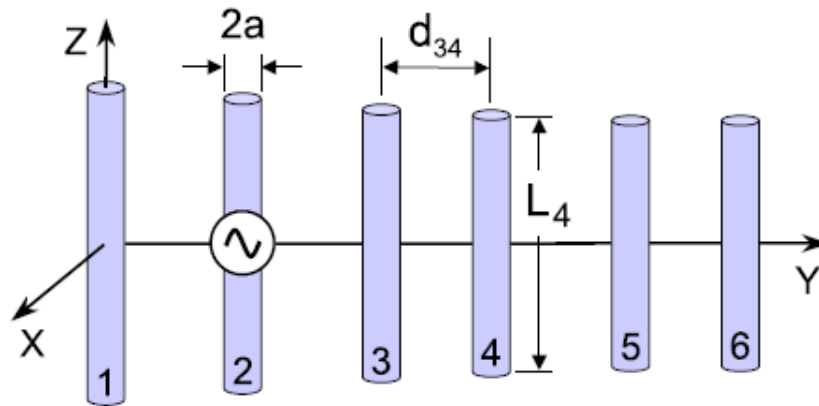


Parâmetros da Antena:

- L_1, L_2, \dots, L_6 = comprimento das hastes;
- $d_{12}, d_{23}, \dots, d_{56}$ = distância entre as hastes;
- $D = 2$ = diâmetro de cada haste (fixo);

Questões Práticas:

- Como calcular o desempenho de transmissão da antena com base nas variáveis?
- Quais os limites para as variáveis?
 - domínio do problema;
- Como calcular o tamanho da antena com base nas variáveis (parâmetros)?



**vetor de variáveis de
otimização**

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \end{bmatrix}$$

$$\mathbf{x}^* = \max_{\mathbf{x}} f(\mathbf{x})$$

← **função objetivo:** desempenho de transmissão

$$\text{sujeito a } \begin{cases} T(\mathbf{x}) \leq T_{max} \\ 0 \leq x_1 \leq 0.5 \\ 0 \leq x_2 \leq 0.5 \\ 0 \leq x_3 \leq 0.5 \\ \vdots \\ 0 \leq x_{11} \leq 0.5 \end{cases}$$

← **restrição:**

- a antena montada não deve exceder o tamanho máximo permitido;

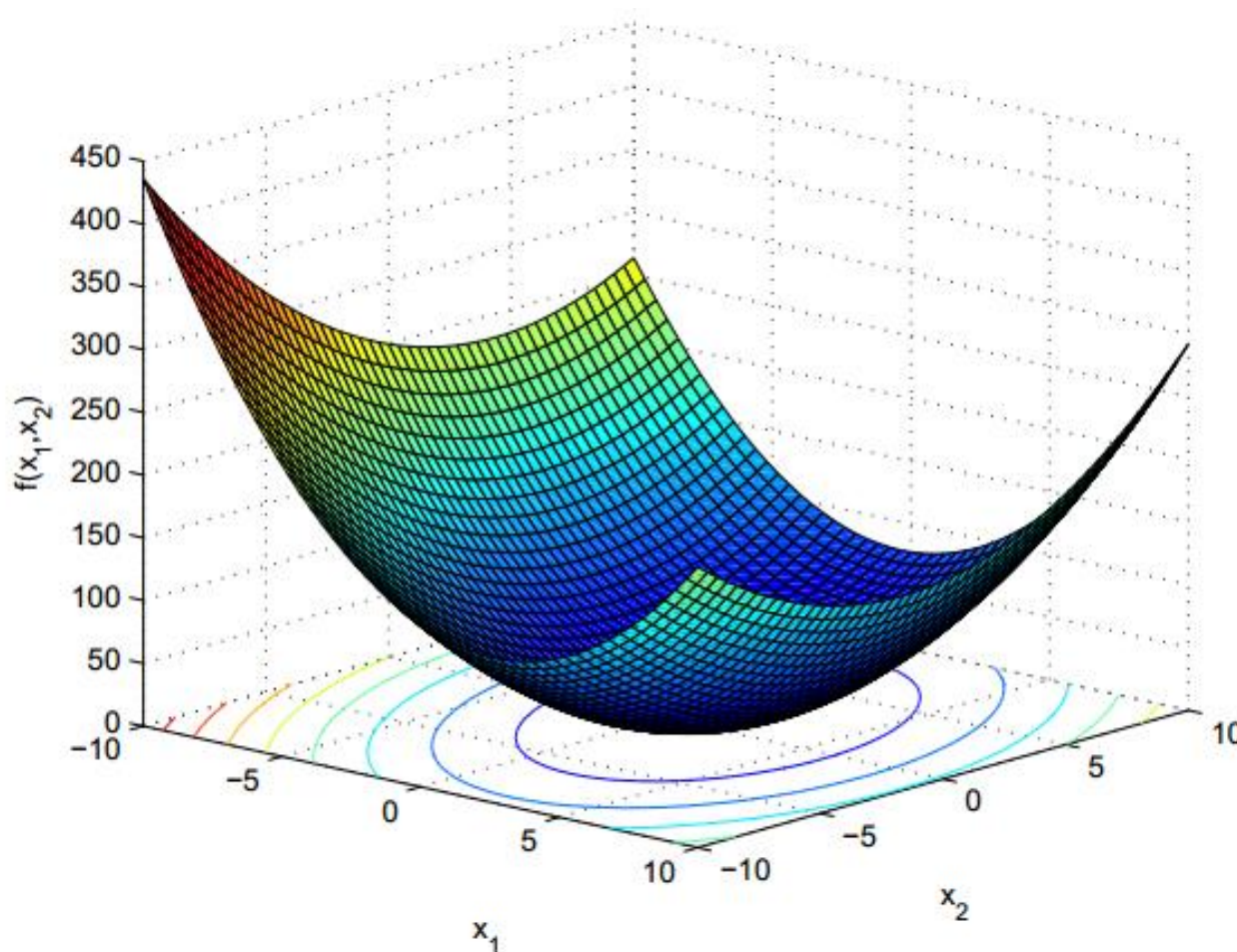
←

restrições de domínio:

- restringem a busca à região do espaço \mathbb{R}^n no qual todas as variáveis tenham valor entre 0 e 0.5m;
- definem o domínio do problema;

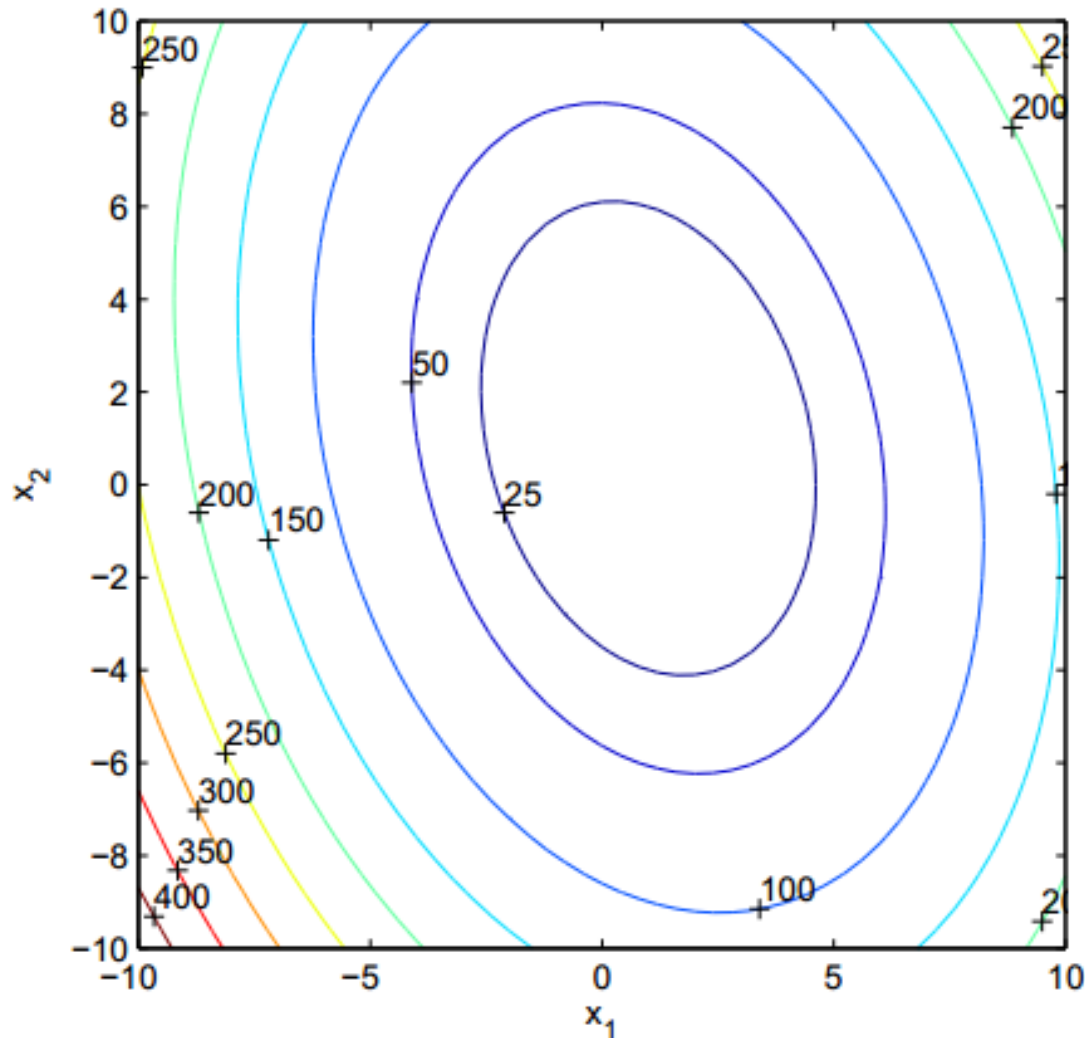
Função Objetivo ($f(\mathbf{x})$)

- consumo de combustível;
- nível de ruído;
- probabilidade de defeitos;
- distância percorrida;

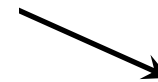


Ex: superfície de $f(\mathbf{x})$
(função quadrática)

Funcao Objetivo ($f(x)$)



- consumo de combustível;
- nível de ruído;
- probabilidade de defeitos;
- distância percorrida;

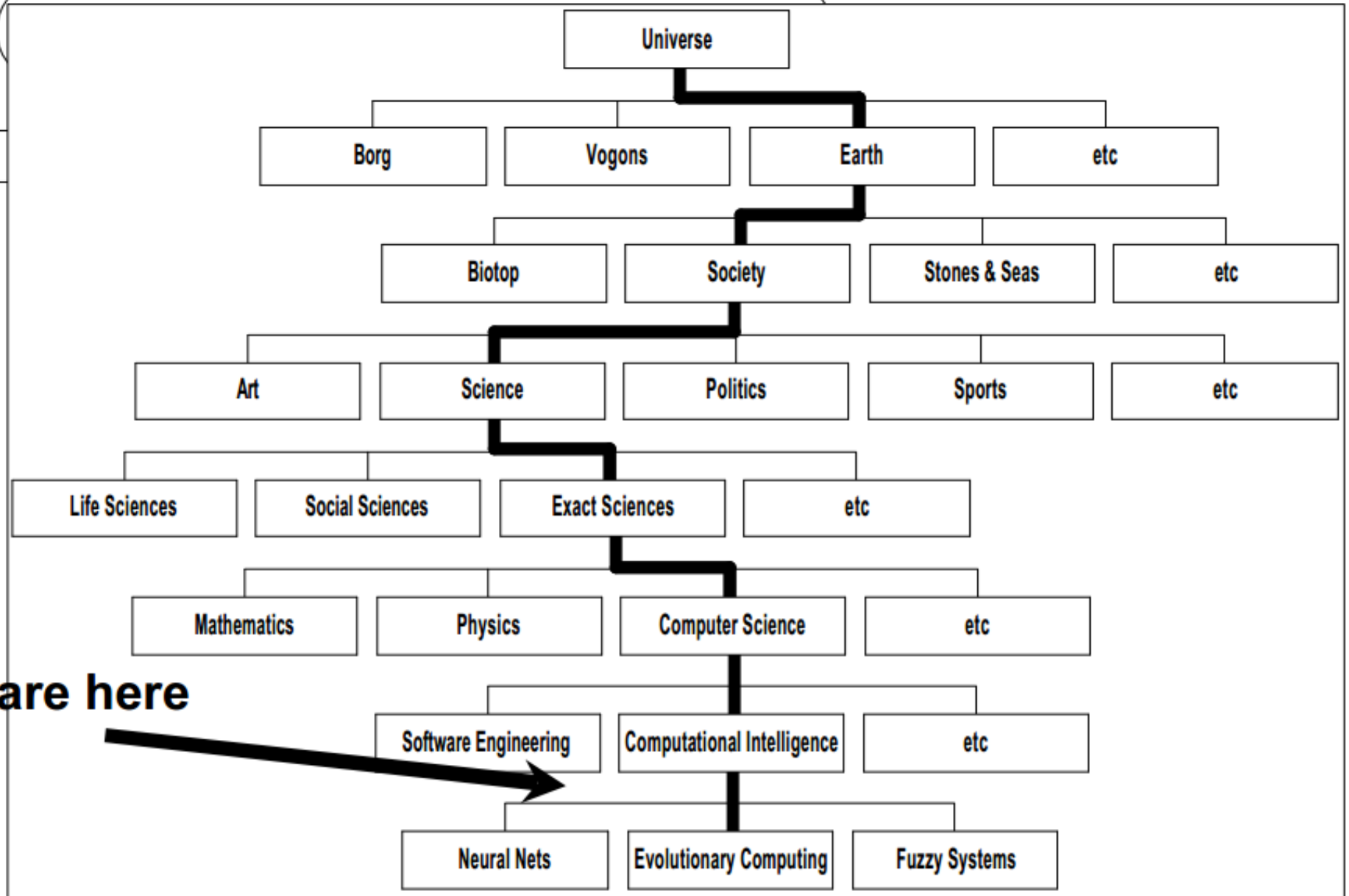


Ex: curvas de nível de $f(x)$
(função quadrática)

COMPUTAÇÃO EVOLUCIONÁRIA (MOTIVAÇÃO)

POSICIONAMENTO DO CURSO:

You are here



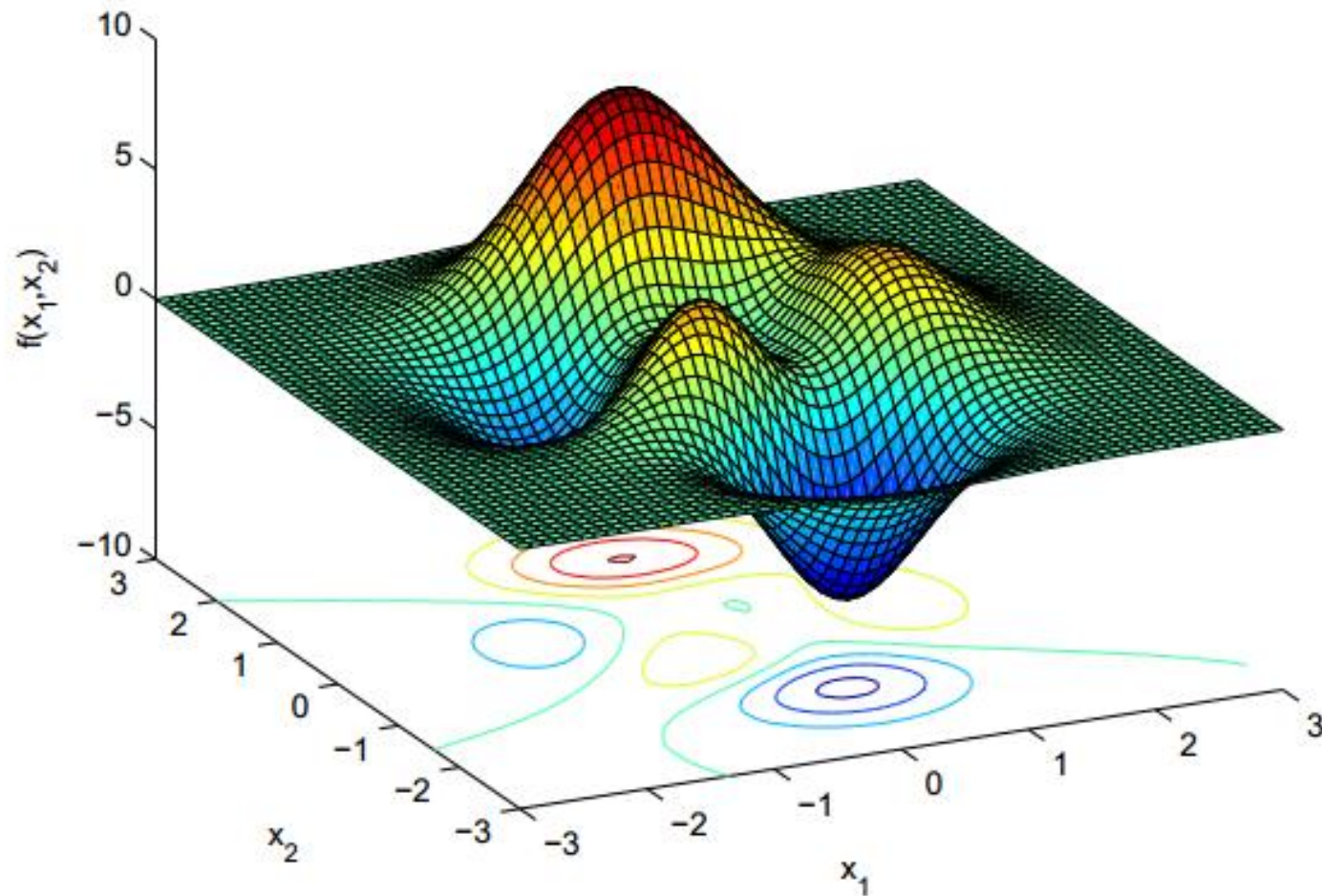
O que é a Computação Evolucionária?

- a **Computação Evolucionária** se refere a um campo da Inteligência Computacional que envolve algoritmos para **otimização** contínua e combinatória de **problemas complexos**;
- os algoritmos da Computação Evolucionária evoluem de forma iterativa (progressiva) uma população de possíveis soluções;
- esta população é selecionada em uma busca aleatória **guiada**, com operadores e processos inspirados por mecanismos biológicos de **Evolução**.

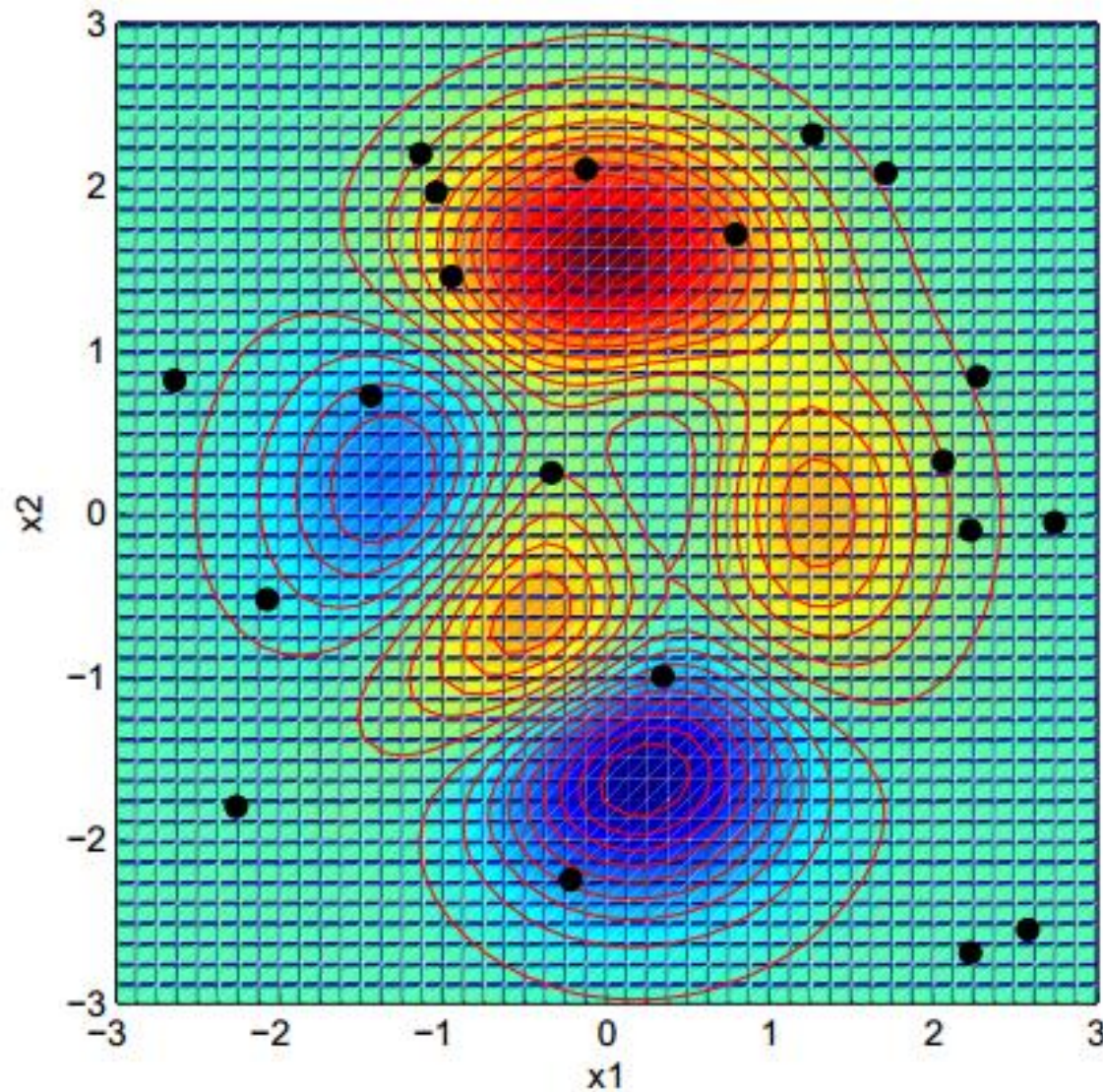
- motivações para o desenvolvimento e aplicação de Algoritmos Evolucionários (AEs):
 - **no contexto de Otimização Contínua:**
 - necessidade de se otimizar funções multimodais ou dinâmicas;
 - **no contexto de Otimização Combinatória:**
 - necessidade de se obter boas soluções para problemas **computacionalmente intratáveis**.

Otimização Contínua

- **Problema:** minimização de uma função objetivo $f(\mathbf{x})$ **multimodal**.

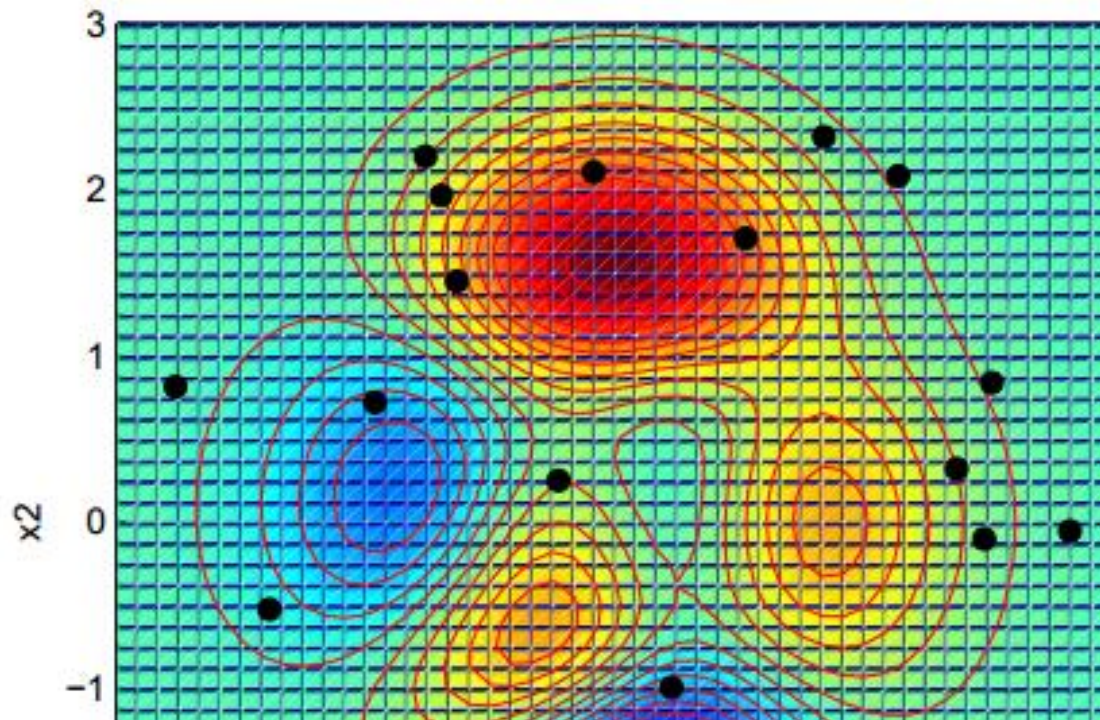


- **Problema:** minimização de uma função objetivo $f(\mathbf{x})$ multimodal



Otimização Contínua

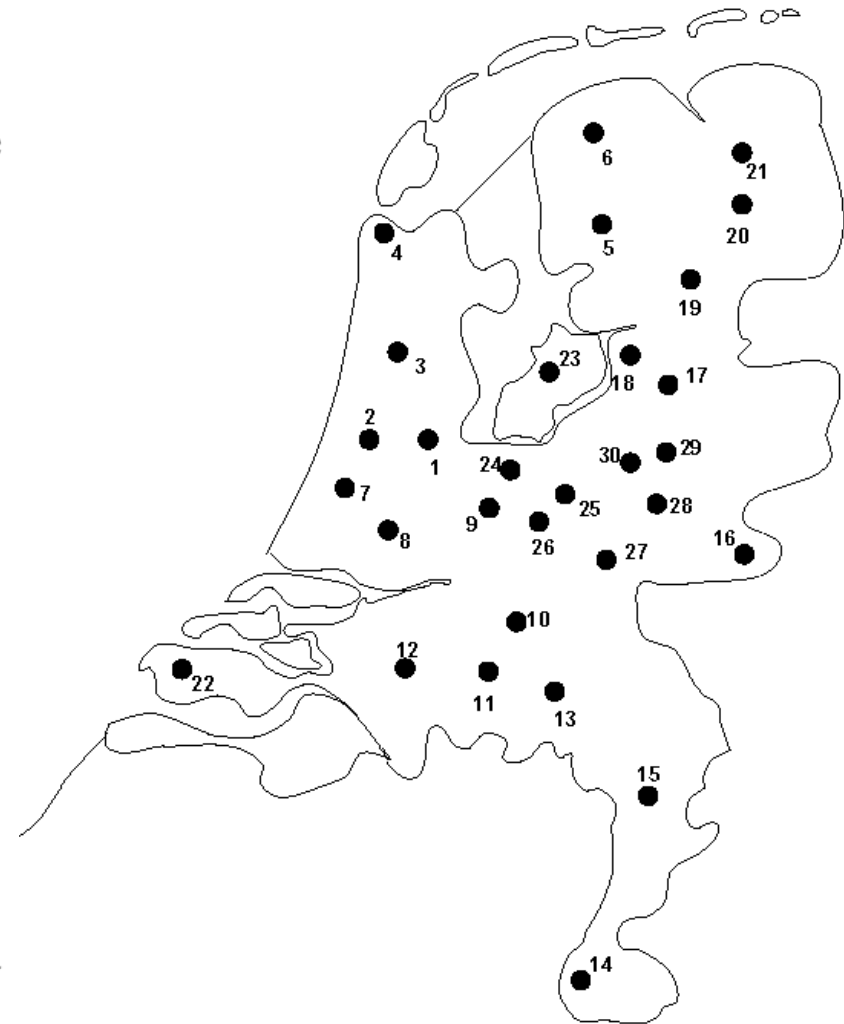
- **Problema:** minimização de uma função objetivo $f(\mathbf{x})$ **multimodal**.



Grande parte das funções objetivo que
queremos otimizar na prática são
multimodais !!!

Otimização Combinatória

- Exemplo 1: *Traveling Salesman Problem* (TSP):
 - o caixeiro deve sair de uma cidade, dita cidade origem, visitar cada uma das $n - 1$ cidades restantes apenas uma única vez e retornar à cidade origem percorrendo a menor distância possível.
 - **Características:**
 - número de rotas possíveis para um problema com n cidades: $(n - 1)!/2$.
 - para 20 cidades, existe $19!/2$ possíveis rotas¹;
 - supondo um computador que analise cada solução em 10ns, seriam necessários cerca de 38 anos para analisar todas as rotas!



1. para o TSP simétrico (o custo de ir é o mesmo de voltar)

Otimização Combinatória

- Exemplo 2: *Cardinality Constrained Portfolio Optimization (CCPO)*:

$$\text{Minimize } \lambda \left[\sum_{i=1}^N \sum_{j=1}^N x_i \cdot x_j \cdot \sigma_{ij} \right] - (1 - \lambda) \left[\sum_{i=1}^N x_i \cdot u_i \right] \quad (4)$$

$$\text{subject to } \sum_{i=1}^N x_i = 1 \quad \begin{array}{l} \text{Risco} \\ \text{Retorno} \end{array} \quad (5)$$

$$\sum_{i=1}^N z_i = K \quad (6)$$

$$\varepsilon_i z_i \leq x_i \leq \delta_i z_i, \quad i = 1, \dots, N \quad (7)$$

$$z_i \in [0, 1], \quad i = 1, \dots, N \quad (8)$$

Otimização Combinatória

- Exemplo 2: *Cardinality Constrained Portfolio Optimization*:
- onde:
 - N = número total de ações;
 - u_i = retorno médio (esperado) de cada ação i ;
 - σ_{ij} = covariância entre os retornos médios das ações i e j ;
 - K = número desejado de ações na carteira (portifolio);
 - x_i = proporção de capital a ser investida na ação i ;
 - ε_i = proporção mínima de capital a ser investida na ação i ;
 - δ_i = proporção máxima de capital a ser investida na ação i ;

Otimização Combinatória

- Exemplo 2: *Cardinality Constrained Portfolio Optimization*:
- **Características:**
 - considerando o problema simplificado em que $x_i \in \{0,1\}$
 - número de carteiras (portifolios) possíveis para um problema com N ações e tamanho de carteria K:
 - $C(N, K) = \frac{n!}{k!(n-k)!}$
 - para $N = 300$ e $K = 10$, existem 1.4×10^{18} possíveis carteiras;
 - supondo um computador que analise cada solução em 10ns, seriam necessários cerca de 44 anos para analisar todas as carteiras.

Otimização Combinatória

- Como encontrar a solução para problemas de natureza combinatória?
 - **Abordagem Força-Bruta (busca exautiva)**: enumere todas as soluções e avalie;
 - Ao se deparar com um problema de otimização combinatória, a seguinte questão deve ser analisada:
 - “Existe algum procedimento computacional que resolva o problema (encontre a solução ótima) em tempo razoável?”
- Tempo Razoável** = procedimento cuja execução demanda um número de operações proporcional a uma expressão polinomial envolvendo o número de variáveis do problema;

Otimização Combinatória

- Muitos problemas de Engenharia são problemas combinatórios para os quais não se conhecem algoritmos capazes de determinar sua solução ótima **garantidamente** em tempo polinomial;
- Tais problemas são computacionalmente intratáveis (NP-Hard), ou seja, seu tempo de solução cresce exponencialmente com o tamanho do problema.

Computação Evolucionária

- Evolutionary Algorithms:
 - Genetic Algorithm; Genetic Programming; Evolutionary Programming; Evolution Strategy; Differential Evolution, etc.
- *Swarm* Intelligence Algorithms:
 - Ant Colony Optimization; Particle Swarm Optimization; Artificial Immune Systems; Cultural Algorithms; Harmony Search; etc.

Computação Evolucionária

- A **Computação Evolucionária** é hoje uma área de pesquisa estabelecida, com dezenas de livros especializados;
- Periódicos Especializados:
 - *IEEE Transactions on Evolutionary Computation*;
 - *MIT Evolutionary Computation*;
 - *Genetic Programming and Evolvable Machines (Springer)*;
 - *Natural Computation (Springer)*;
 - *International Journal of Natural Computing (2010)*;
 - *International Journal of Applied Evolutionary Computation (2010)*;

• Leitura Recomendada:

- CUNHA, TAKAHASHI AND ANTUNES. Manual de Computação Evolutiva e Metaheurística. **(Capítulo 1)**



- A.E. EIBEN, J.E. SMITH. Introduction to Evolutionary Computing (Natural Computing Series), Springer. **(Capítulo 1)**

