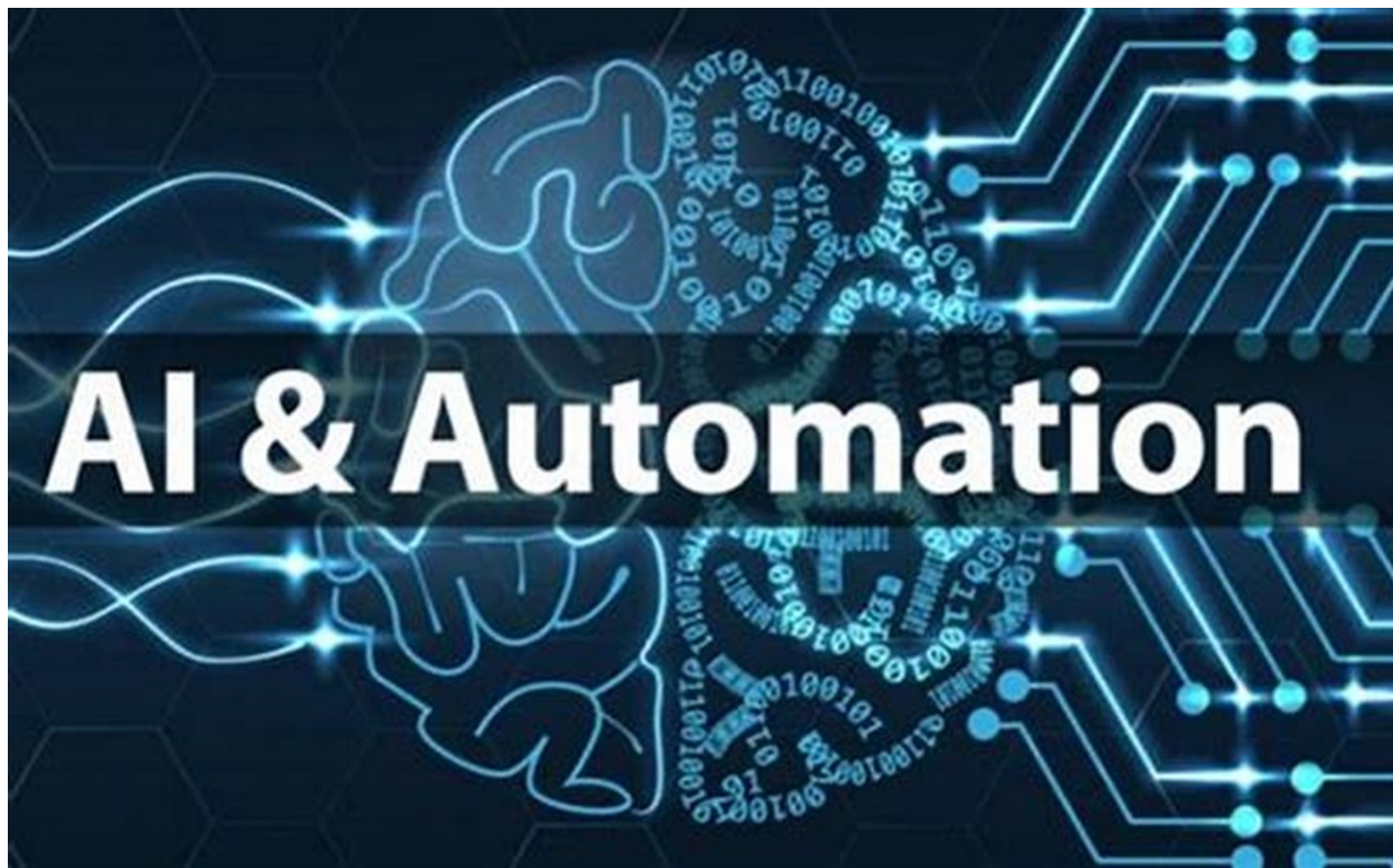


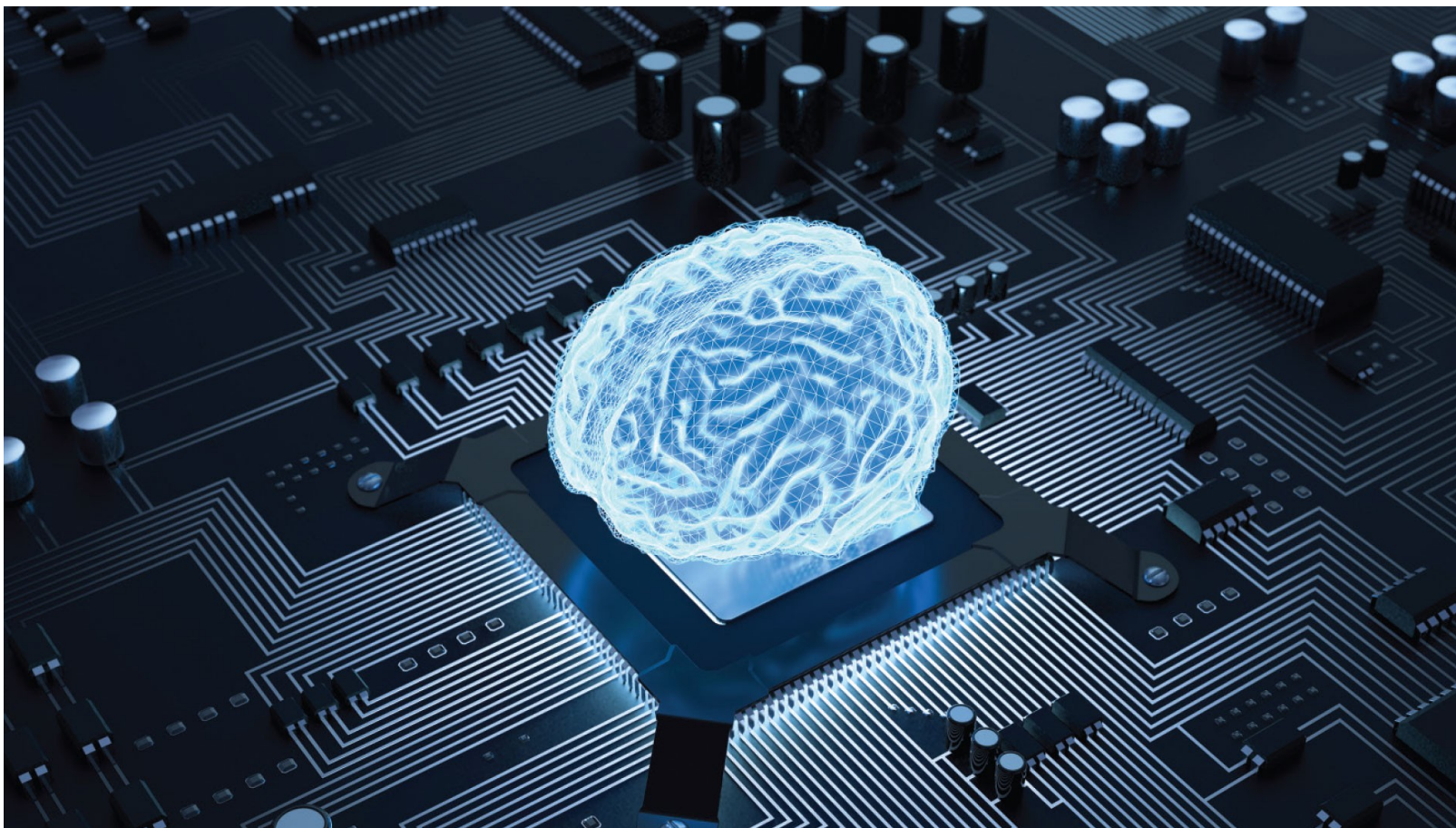
Inteligência Artificial Aplicada à Automação





Inteligência Artificial Aplicada à Automação

8. Algoritmo Genético



Mini-curriculo do Professor

- Pós-doutorado em Ciências (Sistemas Eletrônicos) pela POLI/USP (2023)
- Doutor em Ciências (Sistemas Eletrônicos) pela POLI/USP (2021)
- Mestre em Engenharia Mecânica pela UNISANTA (2017).
- Tutoria EAD pela FGV (2004) e Docência Nível Superior pela FGV (2002).
- MBA Em Gestão Empresarial pela Fundação Getúlio Vargas RJ (2001), com extensão de MBA na Universidade da Califórnia – Campus Irvine (2001).
- Professor Licenciado para ensino de nível segundo grau pelo CEFET – Paraná (1995).
- Engenheiro Eletrônico pela UNISANTA (1991).
- Professor Titular EBTT (2022) da IFSP Cubatão desde 1992. Professor da UNISANTOS (2003 - 2015) e FORTEC (1990 - 1992).
- Pesquisador do EAILab e dos grupos de pesquisa Labmax e AutomSystem do IFSP.
- É colaborador e possui tutoriais publicados no Site Teleco (www.teleco.com.br), desde 2011.
- Obteve Certificações Cisco Business Transformation (2015), PMI (2012), Wireless CWNA, Cisco CCNA & CCNP de Router & Switches (2011).
- Inglês e Espanhol fluentes. Noções de Frances.
- Atuou profissionalmente em todo o Brasil, EUA, Inglaterra, França, Romênia, China e toda LATAM.
- Possui cursos de Fibras Óticas, Microcontroladores, Redes Wireless, Cisco (CCNA, QoS, VoIP), Gerenciamento de Projetos, entre outros.
- Atuou em empresas como Medidata, Cisco, Alcatel-Lucent (Nokia), MSI (hoje Menthum), Evadin , TV Tribuna (Afiliada Rede Globo), ocupando cargos Técnicos, de Consultoria e Gerencia.



Definição:

- **Algoritmo Genético (*genetic algorithm* – GA)** é uma técnica de otimização, baseada nos princípios de Genética e Seleção Natural.
- GA é frequentemente usado para encontrar soluções ótimas ou quase ótimas para problemas difíceis.
- GA pode também ser usado para resolver problemas de otimização, pesquisa e aprendizado automático.
 - Otimização é o processo de fazer algo melhor.
 - Otimização refere-se a encontrar os valores das entradas de forma a obter os "**melhores**" valores de saída.





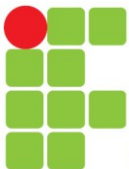
Definição:

- A definição de "**melhor**" varia, porém em geral refere-se a maximizar ou a minimizar uma ou mais funções objetivo (***fitness***), variando os parâmetros de entrada.
- O conjunto de todas as soluções possíveis compõe o **espaço de busca**.
- Um ponto ou um conjunto de pontos no espaço de busca é que dá a solução ideal.
- O objetivo da otimização é encontrar esse ponto ou conjunto de pontos no espaço de busca.
- GAs são algoritmos de pesquisa baseados nos conceitos de seleção natural e genética.
- GAs são um subconjunto da Computação Evolutiva ou evolucionária.



Definição:

- Os GAs foram desenvolvidos por John Holland, David E. Goldberg e seus alunos e colegas da Universidade de Michigan.
- Nos GAs, inicia-se com uma população de possíveis soluções para o problema dado.
- Essas soluções são submetidas a **recombinação e mutação** (como na genética natural), produzindo novas gerações de soluções.
- O processo é repetido por várias gerações.
- Cada indivíduo (ou solução) recebe um **valor de aptidão** (com base no valor da função objetiva).
- Os indivíduos mais aptos têm maior chance de se acasalar.
- Assim, os melhores indivíduos seguem **evoluindo** ou soluções ao longo de gerações, até que se possa atingir um critério de parada.



Vantagens dos GAs:

- São uma poderosa ferramenta para solução de problemas difíceis.
- Não requer nenhuma informação derivativa (que pode não estar disponível para muitos problemas do mundo real).
- É mais rápido e eficiente em comparação com os métodos tradicionais.
- Tem bons recursos para paralelização.
- Otimiza funções contínuas e discretas além de problemas multiobjetivo.
- Fornece uma lista de "boas" soluções e não apenas uma única solução.
- Sempre recebe uma resposta para o problema, que melhora ao longo do tempo.
- Útil quando o espaço de pesquisa é muito grande e há uma grande quantidade de parâmetros envolvidos.



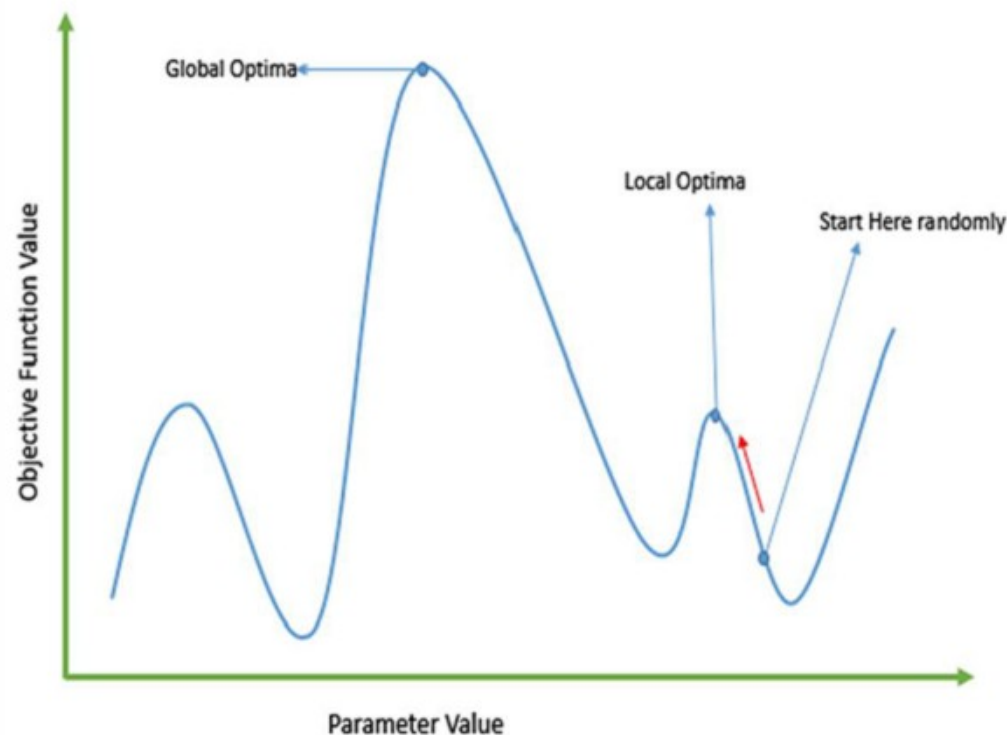
Limitações dos GAs:

- Os GAs não são adequados para todos os problemas, especialmente problemas que são simples e para os quais informações derivativas estão disponíveis.
- O valor de ***Fitness*** é calculado repetidamente, o que pode ser computacionalmente caro para alguns problemas.
- Sendo estocástico, **não há garantias** sobre a otimização ou a qualidade da solução.
- Se não for implementado corretamente, o GA pode não convergir para a solução ideal.



Métodos Baseados em Gradientes:

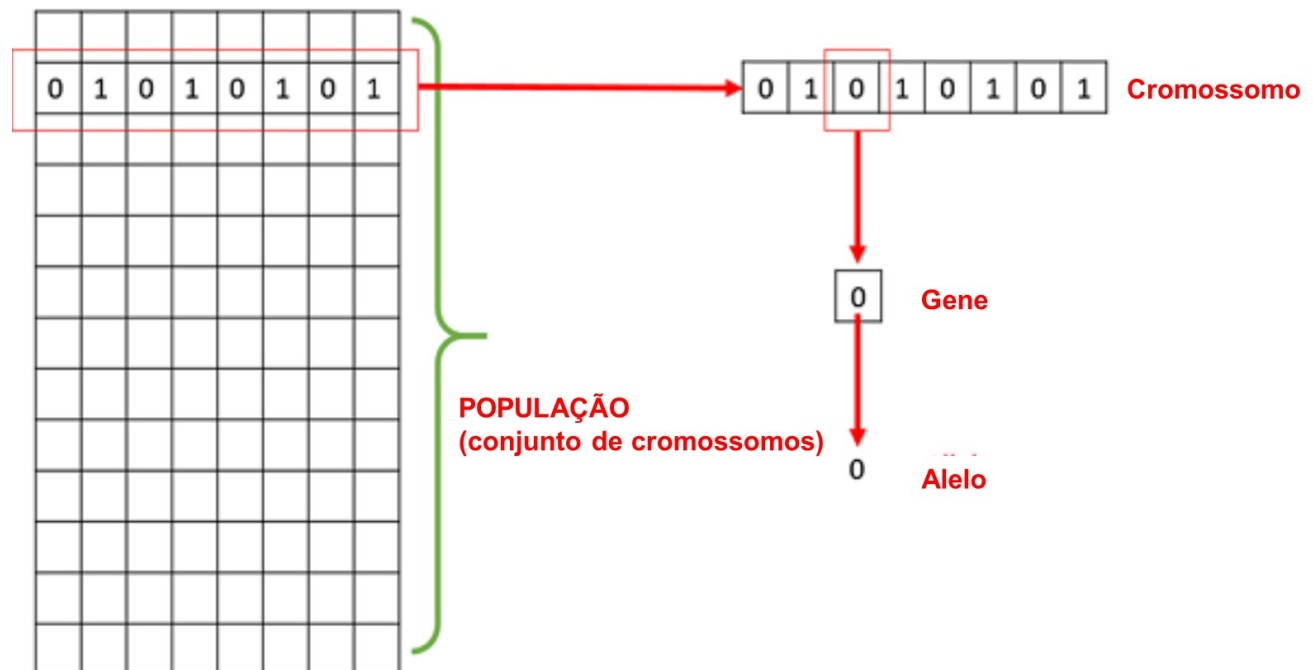
- Os métodos de busca de solução ótima baseados em cálculos tradicionais funcionam começando em um ponto aleatório e movendo-se na direção do gradiente, até chegar ao ponto ótimo.
- Estas técnicas funcionam muito bem para funções objetivos de um único ponto como a função de custo na regressão linear.
- Entretanto, nas situações do mundo real, existem funções feitas de muitos picos e vales, o que faz com que tais métodos falhem.





Terminologia Básica

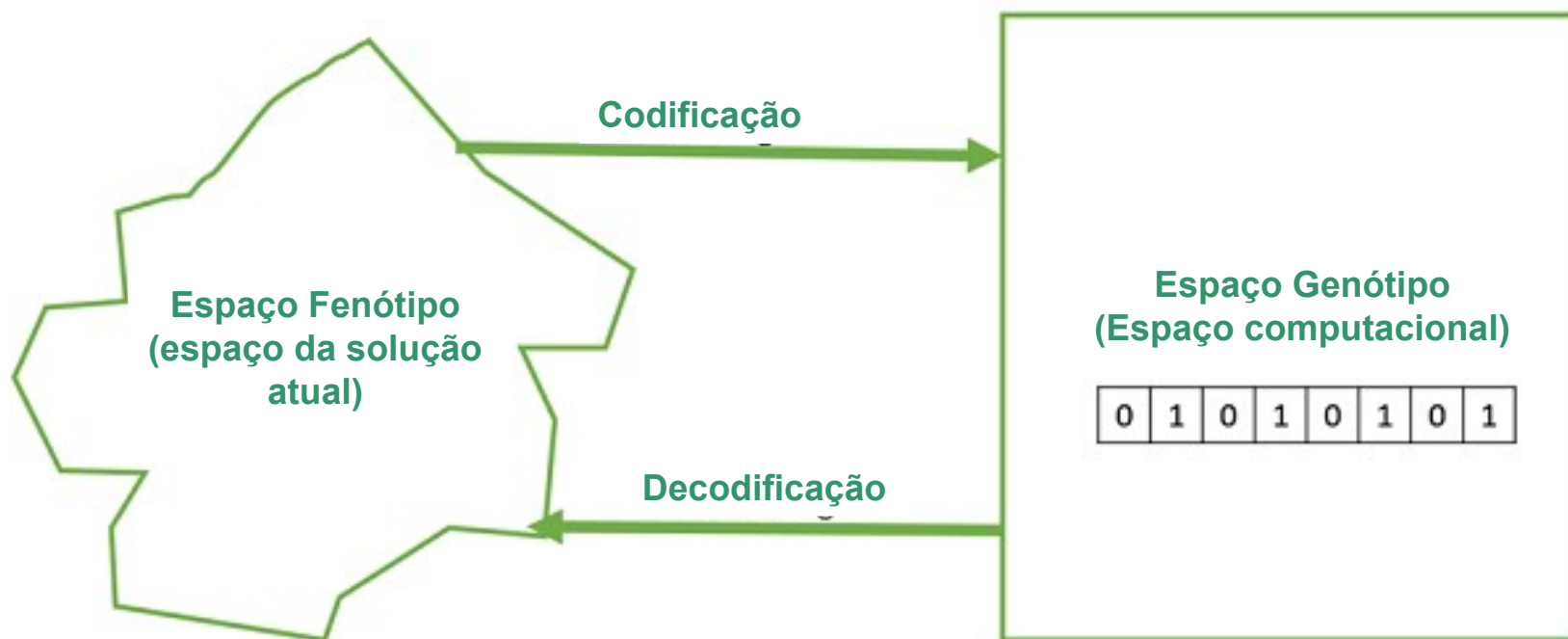
- **População:** subconjunto de todas as possíveis soluções (codificadas) para o problema dado.
- **Cromossomo:** uma das soluções para o problema dado.
- **Gene:** posição ou elemento de um cromossomo.
- **Alelo (*allele*):** valor que um gene leva para um cromossomo em particular.





Terminologia Básica

- **Genótipo:** população no espaço computacional.
- **Fenótipo:** população no espaço real da solução.
- **Decodificação:** processo de transformação de uma solução do genótipo para o espaço do fenótipo.
- **Codificação:** processo de transformação do fenótipo para o espaço de genótipos.

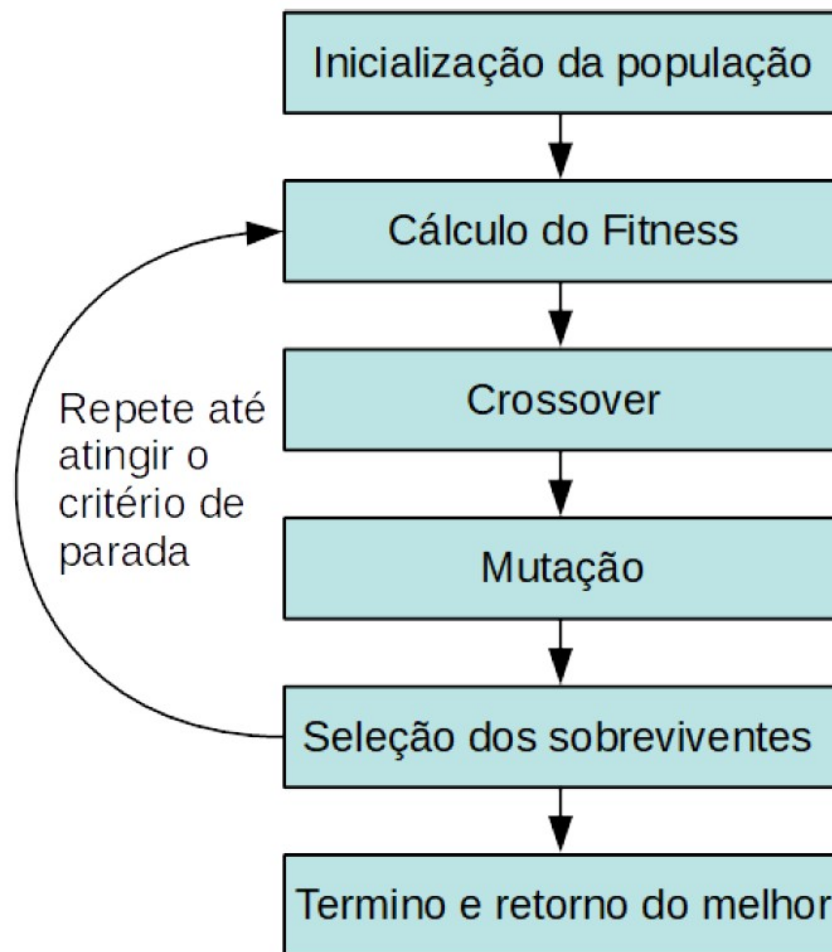




Terminologia Básica

- **Função Objetivo (*sucessor*) ou Função de Fitness:** é uma função que recebe as entradas e retorna a aptidão de um determinado indivíduo.
- **Operadores genéticos:** são operadores que alteram a composição genética da prole. Estes incluem cruzamento, mutação, seleção, etc.

Estrutura Básica de uma GA





Representação do Genótipo

- Binária

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

- Por valores reais

0.5	0.2	0.6	0.8	0.7	0.4	0.3	0.2	0.1	0.9
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

- Por permutação

1	5	9	8	7	4	2	3	6	0
---	---	---	---	---	---	---	---	---	---



Componentes Básicos da GA

- Representação do candidato (população)
 - Importante escolher bem isso. Mais trabalho aqui significa menos trabalho nas funções sucessoras.
- Função(ões) sucessora(s) ou objetivo
 - Mutação, cruzamento (*crossover*)
 - Função Fitness
- Teste de solução
- Alguns parâmetros
 - Tamanho da população
 - Limite de geração



População

- É um subconjunto do conjunto de soluções
- A diversidade da população deve ser mantida, caso contrário poderá levar a uma convergência prematura.
- O tamanho da população não deve ser mantido muito grande, pois vai exigir muito poder computacional.
- Uma população muito pequena pode não ser suficiente para um bom conjunto de acasalamento.
 - Portanto, um tamanho ideal de população precisa ser definido por tentativa e erro.
- Normalmente a população é definida como uma matriz de duas dimensões



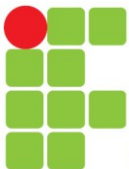
Inicialização da População

- **Inicialização aleatória:**
 - Preencha a população inicial com soluções completamente aleatórias.
- **Inicialização heurística:**
 - Preencha a população inicial usando uma heurística conhecida para o problema.
- A literatura aponta que a inicialização heurística, em alguns casos, afeta apenas a aptidão inicial da população, mas, no final, é a diversidade das soluções que levam à otimização.



Modelos de População

- **Estado estacionário:**
 - Gera-se uma ou duas proles em cada iteração e elas substituem um ou dois indivíduos da população.
 - Um GA de estado estacionário também é conhecido como GA Incremental.
- **Geracional:**
 - Gera-se 'n' indivíduos, onde n é o tamanho da população, e toda a população é substituída pela nova no final da iteração.



Função Objetivo

- É uma função que toma uma solução candidata como entrada e produz como saída um valor que representa o quão “boa” a solução é em relação ao problema em consideração.
- O cálculo do valor da aptidão é feito repetidamente em um GA e, portanto, deve ser rápido para evitar sobrecargas de processamento.
- Em alguns casos, calcular a função objetivo pode não ser possível devido a complexidades do problema.
 - Nesses casos, faz-se uma aproximação as aptidão para atender necessidades do GA.



Função Objetivo

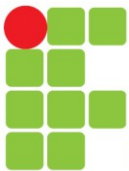
- **Mutação:** Dado um candidato, retorne um candidato ligeiramente diferente.
- **Cruzamento (Crossover):** dos dois candidatos, produza um que contenha elementos de cada um.
- Nem sempre gera-se um sucessor para cada candidato. Em vez disso, gera-se uma população sucessora com base nos candidatos da população atual, ponderada pela aptidão.



Função Objetivo

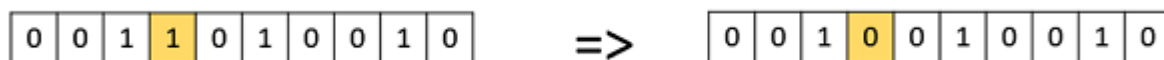
- **Mutação:**

- Em termos simples, a mutação pode ser definida como um pequeno ajuste aleatório no cromossomo, para obter uma nova solução.
- A mutação é usada para manter e introduzir diversidade na população e é geralmente aplicado com baixa probabilidade.
- Se a probabilidade é muito alta, o GA é reduzido a uma pesquisa aleatória.
- Mutação é a parte do GA que está relacionada a “exploração” do espaço de busca.
- Foi observado que a mutação é essencial para a convergência do GA, enquanto o crossover não é.



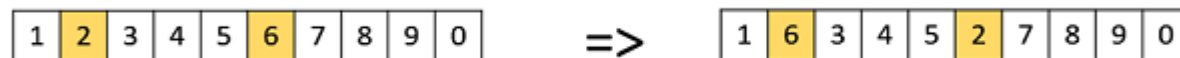
Operadores de Mutação

- Inversão de bit

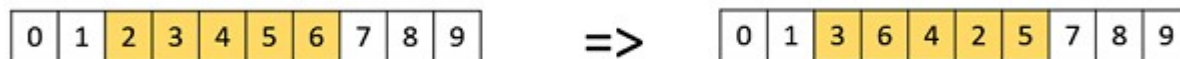


Recomposição aleatória:

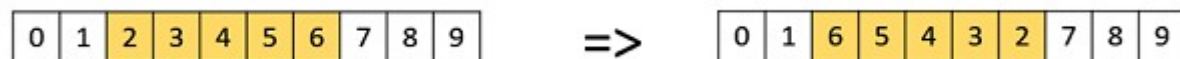
- Mutação por troca



- Mutação por embaralhamento



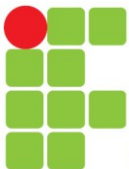
- Mutação por inversão



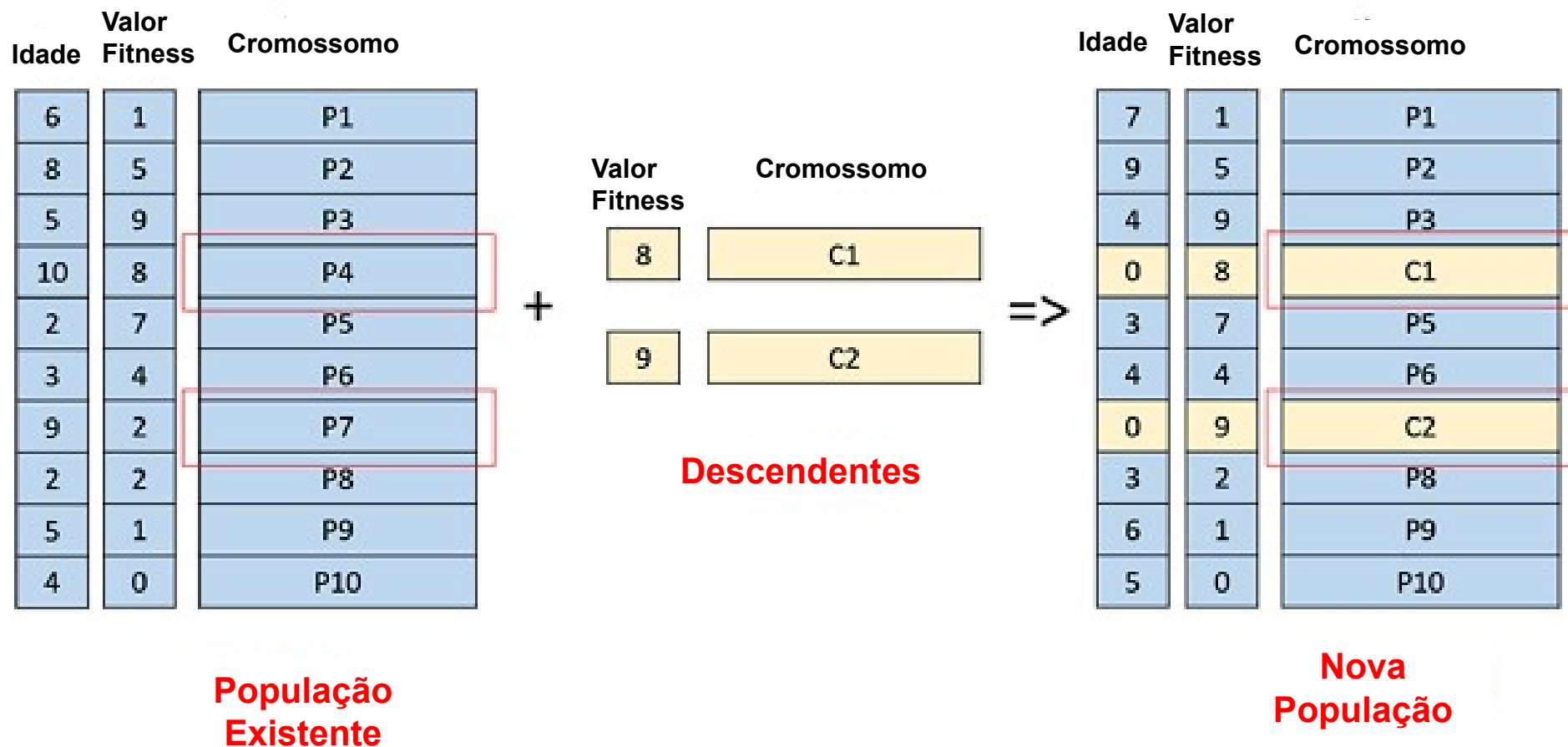


Seleção dos Sobreviventes

- A Política de Seleção de Sobreviventes determina quais indivíduos devem ser expulsos e quais devem ser mantidos na próxima geração.
- É crucial, pois deve garantir que os indivíduos mais aptos não sejam expulsos da população, enquanto, ao mesmo tempo, a diversidade deve ser mantida.
- Alguns GAs empregam o elitismo. Isso significa que o atual membro mais apto da população é sempre propagado para a próxima geração.
 - Assim, sob nenhuma circunstância o membro mais apto da população atual pode ser substituído.
- A política mais fácil é expulsar membros aleatórios da população, mas essa abordagem frequentemente tem problemas de convergência, portanto, as estratégias a seguir são amplamente usadas:
 - Seleção baseada na idade
 - Seleção baseada no fitness

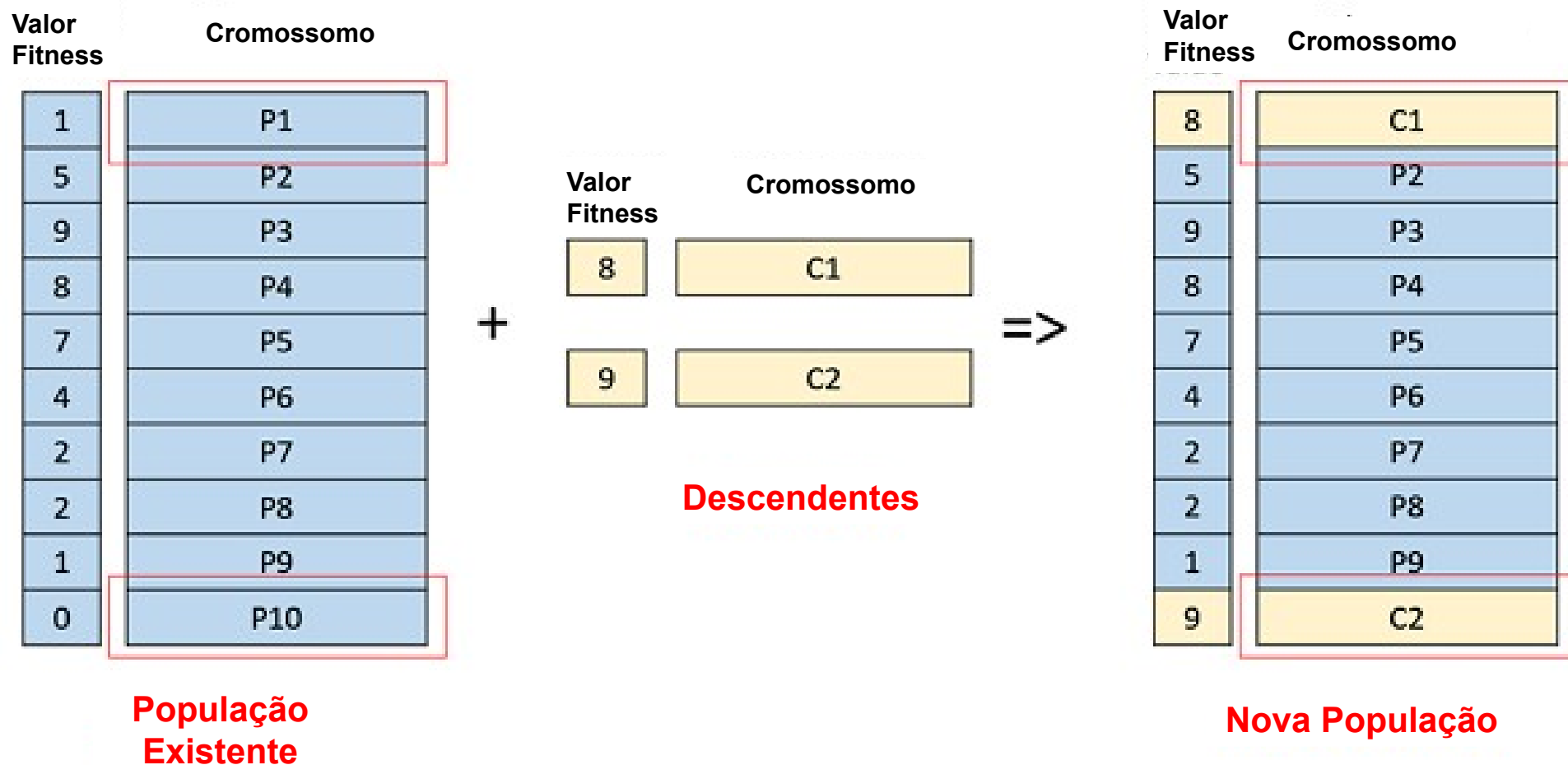


Seleção baseada na idade





Seleção baseada no Fitness





Critério de Parada

- É importante para determinar quando a execução do GA terminará.
- Inicialmente o GA progride muito rapidamente, encontrando melhores soluções a cada poucas iterações, mas isso tende a saturar nos estágios posteriores.
- Assim, um critério de parada é necessário para definir o ponto final da execução.
- Normalmente, mantem-se um dos seguintes critérios de parada.
 - Quando não houve melhora na população por X iterações.
 - Quando alcança-se um número absoluto de gerações.
 - Quando o valor da função objetivo atingiu um determinado valor pré-definido.



Aplicações dos GAs:

- Otimização de projetos de engenharia, como aerodinâmica, eficiência de turbinas eólicas, etc.
- Programação e Logística, como otimização de rotas, entregas, programação de vôos, etc.
- Biotecnologia e Farmacologia, como na identificação e combinação de compostos para doenças específicas.
- Jogos e Simulações.
- Mercado financeiro, como para otimização de estratégias de trading.
- Reconhecimento de padrões e aprendizado de máquina (*machine learning* – ML).

Exemplo de código GA em Java:

<https://medium.com/rapaduratech/introdução-aos-algoritmos-genéticos-405825c1e281>

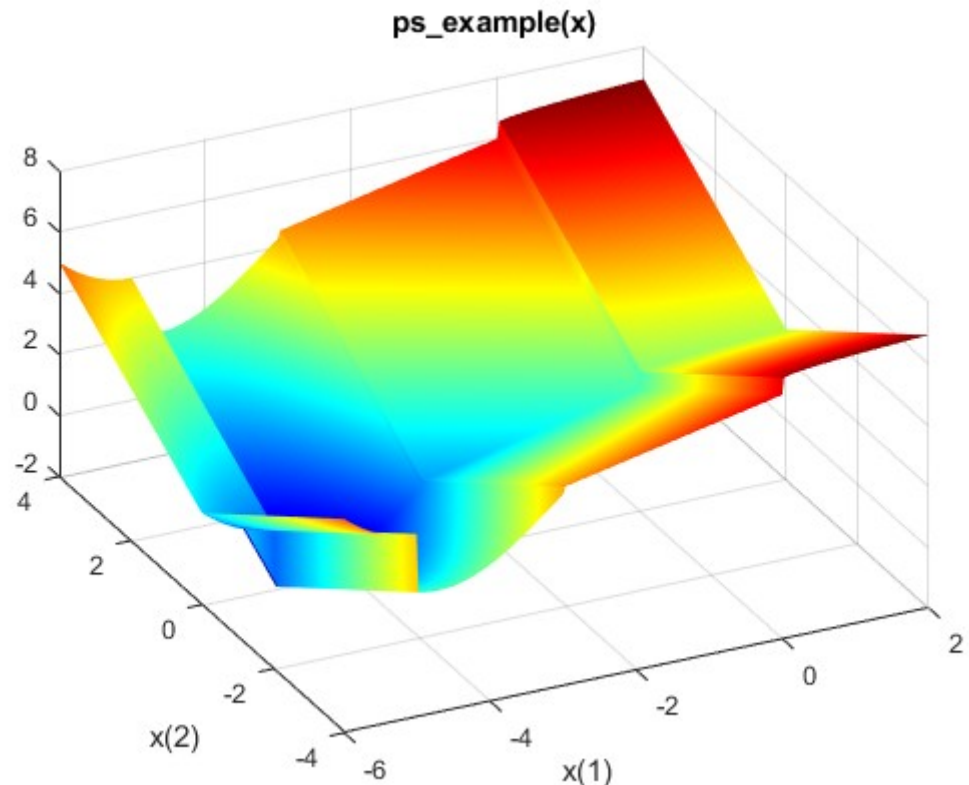


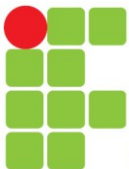
Exemplos em Matlab

1. Otimize uma função não suave (*nonsmooth*) usando GA

a. Plot a função:

```
xi = linspace(-6,2,300);  
yi = linspace(-4,4,300);  
[X,Y] = meshgrid(xi,yi);  
Z = ps_example([X(:),Y(:)]);  
Z = reshape(Z,size(X));  
surf(X,Y,Z,'MeshStyle','none')  
colormap 'jet'  
view(-26,43)  
xlabel('x(1)')  
ylabel('x(2)')  
title('ps\_example(x)')
```





Exemplos em Matlab

1. Otimize uma função não suave (*nonsmooth*) usando GA

b. Encontre o mínimo da função usando GA:

```
rng default % For reproducibility  
x = ga(@ps_example,2)
```

Optimization terminated: average change in the fitness value less than options.
FunctionTolerance.

x =

-4.6793 -0.0860



Exemplos em Matlab

2. Use o GA para minimizar a função ps_example na região:

e

Usando uma restrição de tolerância que é menor que o default.

a) Converta as 2 restrições em matriz

```
A = [-1 -1];  
b = -1;  
Aeq = [-1 1];  
beq = 5;
```

b) Para obter uma solução mais precisa, defina uma tolerância de restrição de $1e-6$. E para monitorar o progresso do solucionador, defina uma função de plotagem.

```
options = optimoptions('ga','ConstraintTolerance',1e-6,'PlotFcn', @gaplotbestf);
```



Exemplos em Matlab

2. Use o GA para minimizar a função ps_example na região:

c) Resolva o problema de minimização

```
rng default % For reproducibility  
fun = @ps_example;  
x = ga(fun,2,A,b,Aeq,beq,[],[],[],options)
```

Optimization terminated: average change in the fitness value less than options. Function Tolerance.

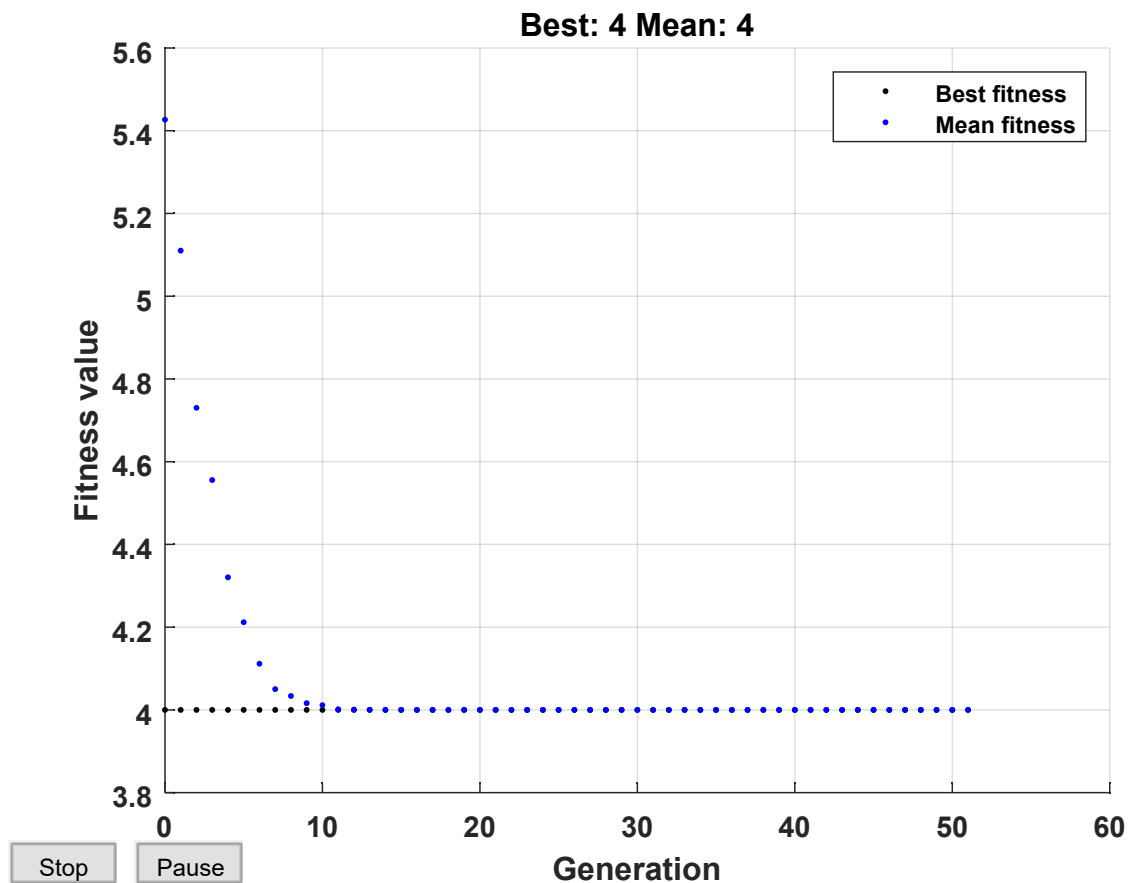
x =

-2.0000 3.0000



Exemplos em Matlab

2. Use o GA para minimizar a função `ps_example` na região:





Exemplos em Matlab

2. Use o GA para minimizar a função ps_example na região:

d) Verifique se as restrições lineares são satisfeitas dentro de $1e^{-6}$.

```
>> disp(A*x' - b)  
9.9999e-07
```

```
>> disp(Aeq*x' - beq)  
-9.8695e-07
```

- Bringsjord, Selmer and Naveen Sundar Govindarajulu, "Artificial Intelligence", The Stanford Encyclopedia of Philosophy (Summer 2024 Edition), Edward N. Zalta & Uri Nodelman (eds.) Disponível em: <https://plato.stanford.edu/archives/sum2024/entries/artificial-intelligence/>. Acessado em Maio 5, 2024.
- RUSSELL, Stuart J.; NORVIG, Peter. *Artificial intelligence: a modern approach*. Pearson, 4th edition, 2016.
- Kerschbaumer, R. Algoritmos Genéticos - Tópicos m Inteligência Artificial, Instituto Federal Catarinense, Campus Luzerna. Disponível em: <https://ricardokers.github.io/Anexos/Slides%20Algoritmos%20Genéticos.pdf>. Acessado em Maio 10, 2024.
- DE OLIVEIRA, A. M.; JUSTO, J. F.; BARROQUEIRO, C. H.; CARVALHO, A. AUTOMATO CELULAR GÊNESIS: Um Ambiente Virtual Baseado Em Autômato Celular Para Simulação de Vida Artificial. Revista Acadêmica - Ensino de Ciências e Tecnologias do IFSP – Campus Cubatão, QUALIF, ISSN 2595-2277, VOLUME 3 - NÚMERO 3 – AGOSTO/DEZEMBRO DE 2018, pag. 72-90. Disponível em: https://intranet.cbt.ifsp.edu.br/qualif/volume03/ARTIGO_05_03.pdf. Acessado em Maio 13, 2024.
- Help Center. Find minimum of function using genetic algorithm. Mathworks. Disponível em: <https://www.mathworks.com/help/gads/ga.html>. Acessado em Maio 13, 2024.
- NORONHA, O. Introdução aos Algoritmos Genéticos com Java. Rapadura Tech. Medium. Set 4, 2023. Disponível em: <https://medium.com/rapaduratech/introdução-aos-algoritmos-genéticos-405825c1e281>. Acessado em Maio 13, 2024.
- PACHECO, Marco Aurélio Cavalcanti et al. Algoritmos genéticos: princípios e aplicações. ICA: Laboratório de Inteligência Computacional Aplicada. Departamento de Engenharia Elétrica. Pontifícia Universidade Católica do Rio de Janeiro. Fonte desconhecida, v. 28, 1999. Disponível em: https://www.inf.ufsc.br/~mauro.roisenberg/ine5377/Cursos-ICA/CE-intro_apost.pdf. Acessado em Maio 13, 2024.

Para Finalizar



Perguntas?



adecarvalhojr@ifsp.edu.br