

# INF623

2024/1



# Inteligência Artificial

## A5: Busca local e otimização II

# Plano de aula

- ▶ Algoritmos genéticos
  - ▶ Representação de candidatos
  - ▶ Função de adaptação
  - ▶ Geração de vizinhos por reprodução
    - ▶ Cruzamento
    - ▶ Mutação
  - ▶ Aplicações

# Algoritmos Genéticos

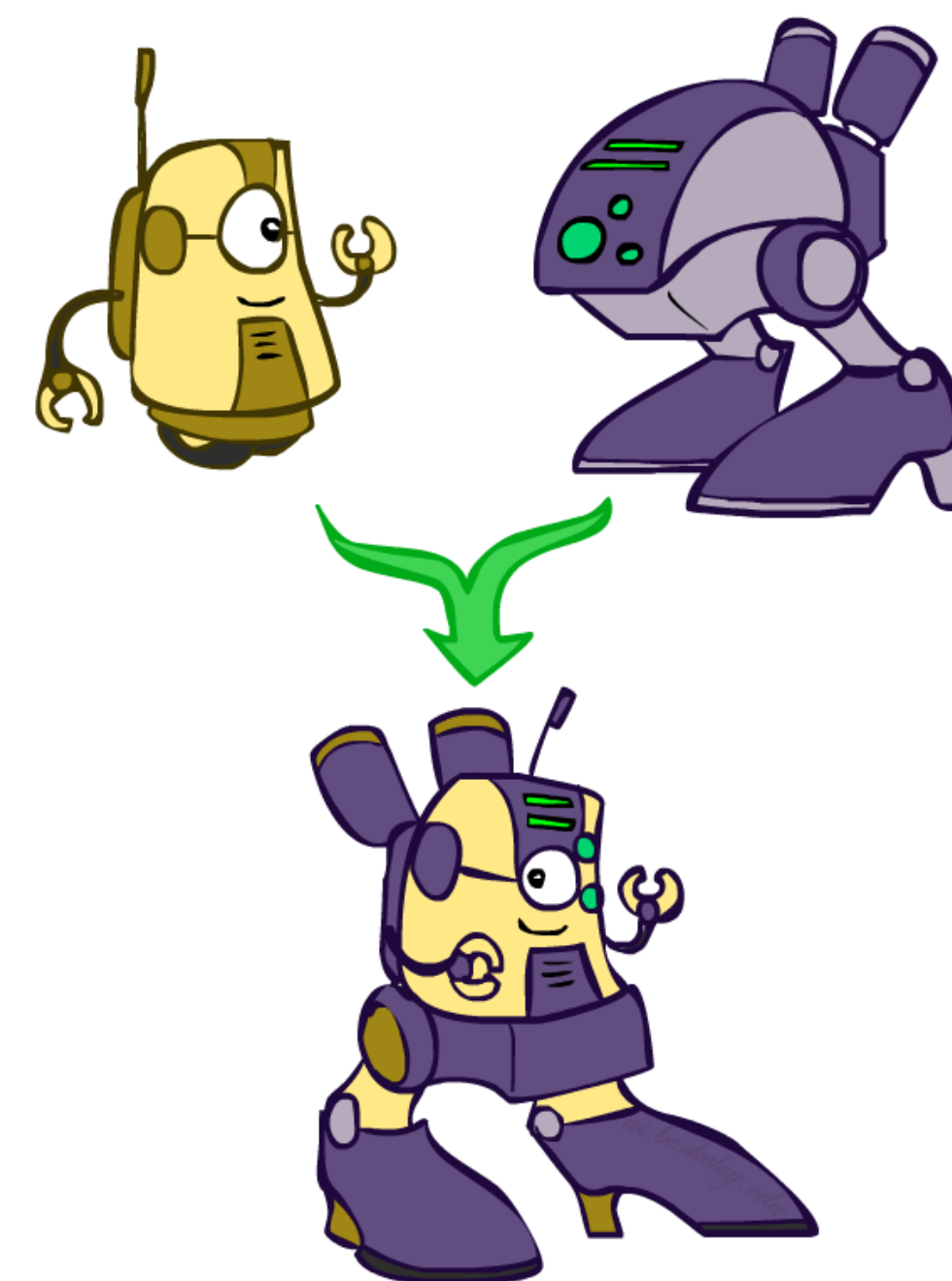
**Algoritmos genéticos** são algoritmos de busca local inspirados na teoria da seleção natural da Biologia.

- ▶ **População** – assim como na busca em feixe, mantém um conjunto de soluções candidatas
- ▶ **Função de adaptação** – função objetivo ou heurística
- ▶ **Seleção** – vizinhança é criada a partir de um subconjunto de candidatos promissores
- ▶ **Reprodução** – candidatos selecionados produzem vizinhos aleatoriamente de duas formas
  - ▶ Cruzamento (combinação de duas soluções)
  - ▶ Mutação (alteração pequena em uma solução)

# Algoritmos Genéticos

Seja **m** o tamanho da população, **n** o número de gerações e **v** a função de adaptação:

```
def algoritmos-geneticos(m, n, v):  
    1. populacao = init_populacao(m)  
    2. for i in n:  
        3. pool = seleção(populacao, v)  
        4. populacao = cruzamento(pool)  
        5. mutação(populacao)  
    7. return argmax(populacao)
```



# Inicialização da população

Para inicializar a população, nós precisamos:

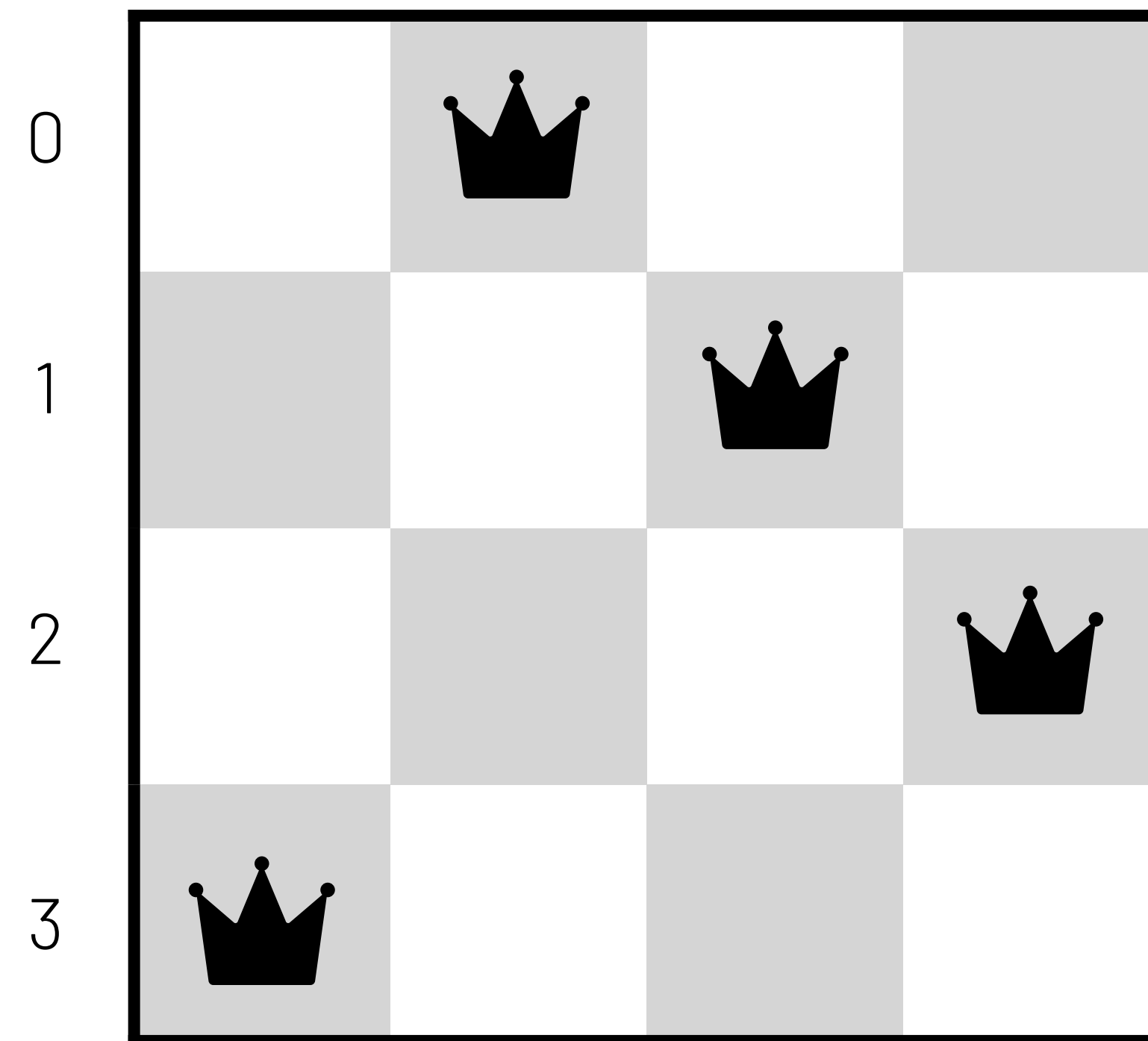
1. Definir uma representação (**genótipo**) para as soluções candidatas (**indivíduos**)

Tradicionalmente definida na forma de arranjos unidimensionais, por exemplo:

▶ Binária	0	0	1	0	1	→ <b>Representação clássica</b>
▶ Inteira	2	2	0	1	-2	
▶ Real	0,2	-3,1	7,4	0,09	1.0	

2. Gerar aleatoriamente um conjunto com M

# Exemplos de representação de n-rainhas



Podemos representar o problema das N-rainhas tanto por um arranjo de inteiros como

► Binária (tabuleiro achatado)

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

► Inteira (posição das rainhas)

3	0	1	2
---	---	---	---

# Função de aptidão (fitness)

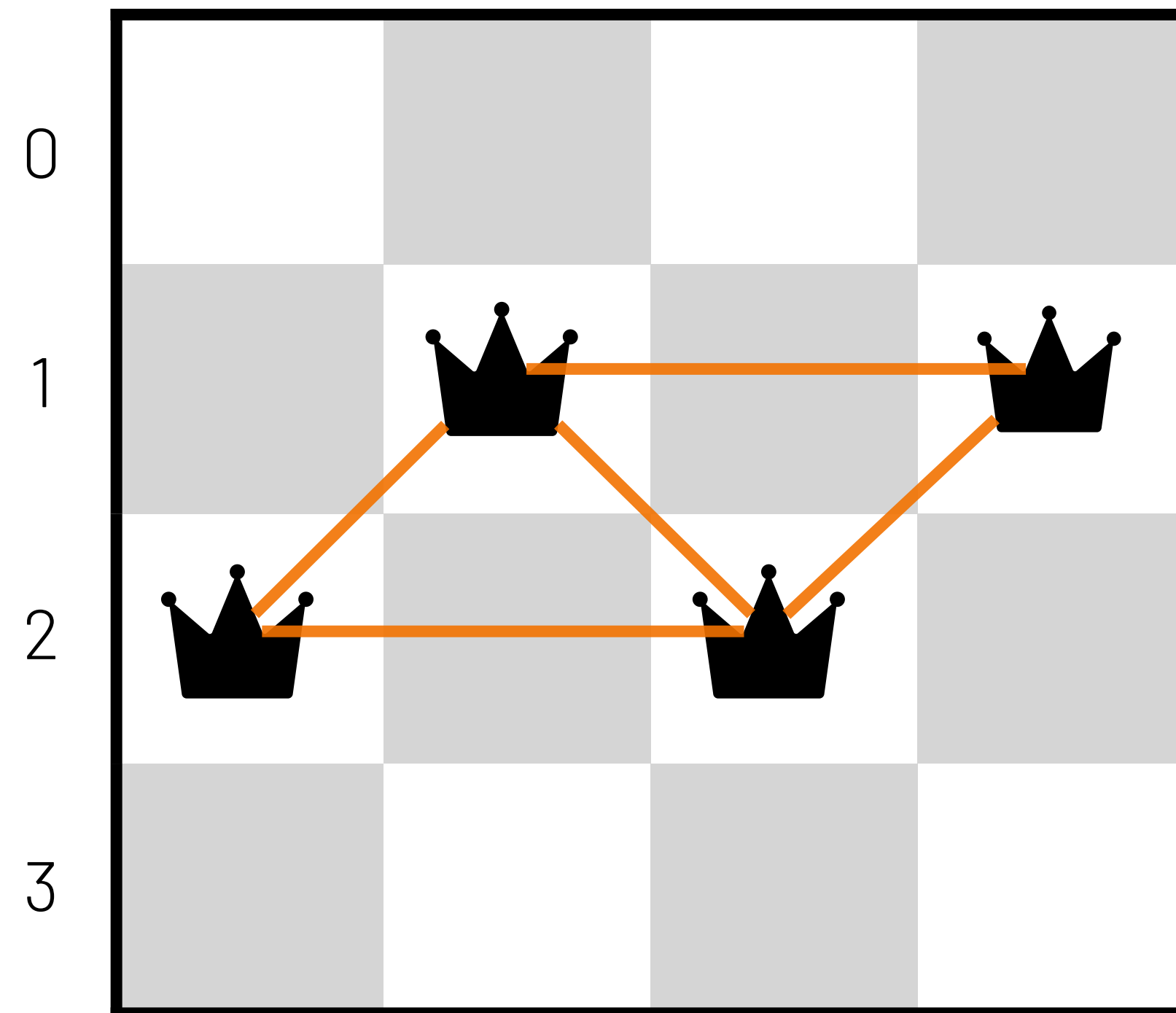
A **função de aptidão** é uma função  $v : S \rightarrow \mathbb{R}$  que recebe um indivíduo  $c \in C$  e retorna a qualidade desse indivíduo.

Diferentes tipos de função de aptidão foram propostas com AGs, exemplos:

- ▶ Diretas
- ▶ Baseadas em simulação
- ▶ Interativas

# Função de aptidão diretas

A qualidade dos genótipos é calculada diretamente por uma função matemática.



Problema das N-rainhas

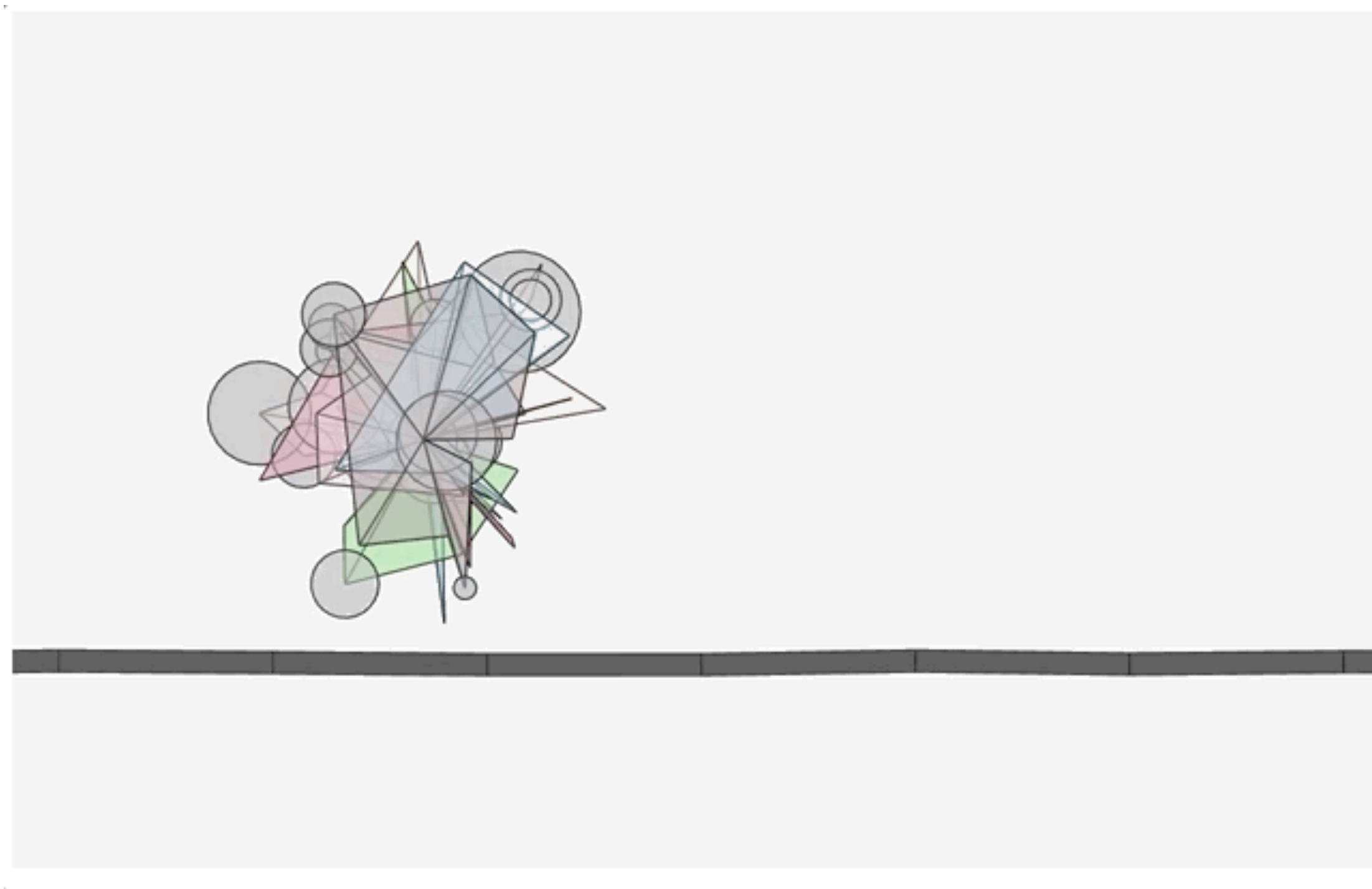
**Número de rainhas se atacando**

$$v\left(\begin{array}{|c|c|c|c|}\hline 2 & 1 & 2 & 1 \\\hline\end{array}\right) = 5$$



# Função de aptidão baseadas em simulação

A qualidade dos genótipos é definida como resultado de uma simulação.



## BoxCar2D

Um algoritmo genético para evoluir formas aleatórias com duas rodas em carros ao longo de gerações. A distância percorrida pelo carro define sua qualidade.

# Função de aptidão interativas

A qualidade dos genótipos é definida por avaliadores humanos



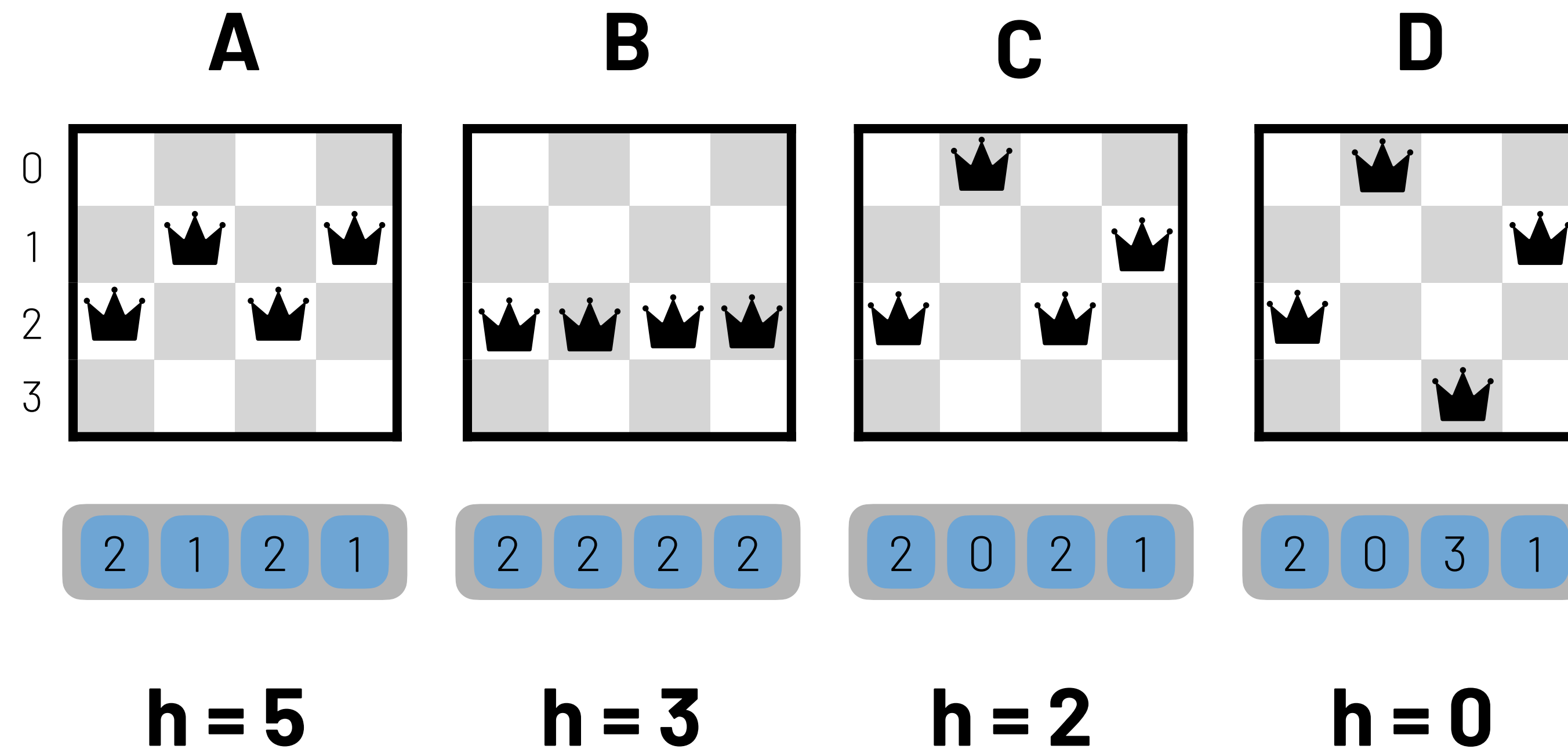
Karl Sims, Galápagos, 1997

Galápagos is an interactive Darwinian evolution of virtual "organisms." Twelve computers simulate the growth and behaviors of a population of abstract animated forms and display them on twelve screens arranged in an arc. The viewers participate in this exhibit by selecting which organisms they find most aesthetically interesting and standing on step sensors in front of those displays. The selected organisms survive, mate, mutate and reproduce.

# Seleção

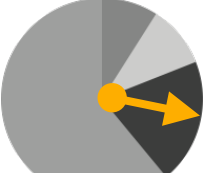


A etapa de **seleção** produz uma lista de pais, chamada *grupo de acasalamento*, para reproduzir. Existem três métodos comuns para criar esse grupo:

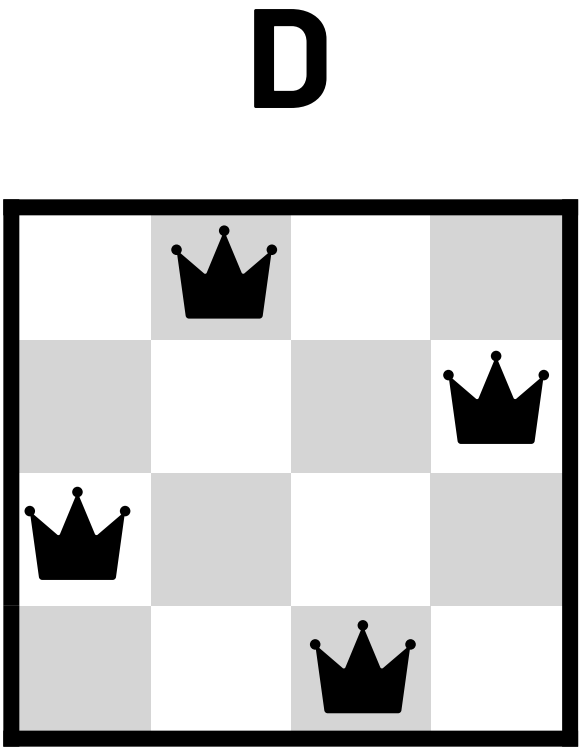
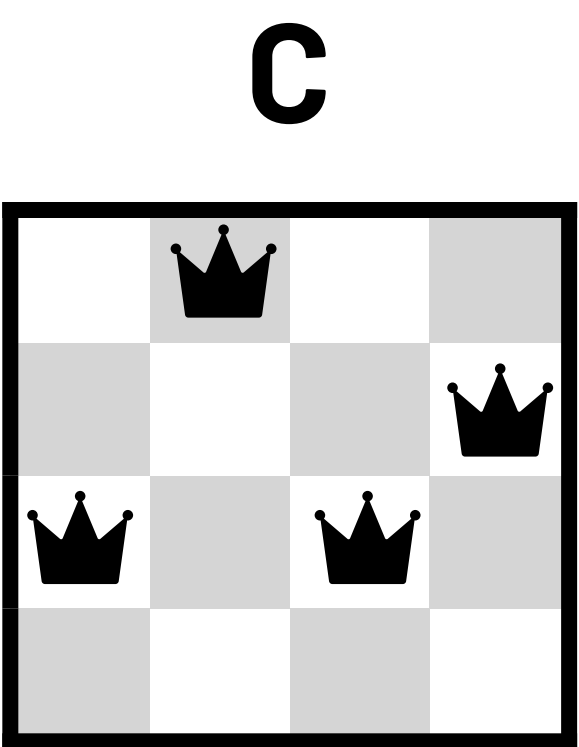
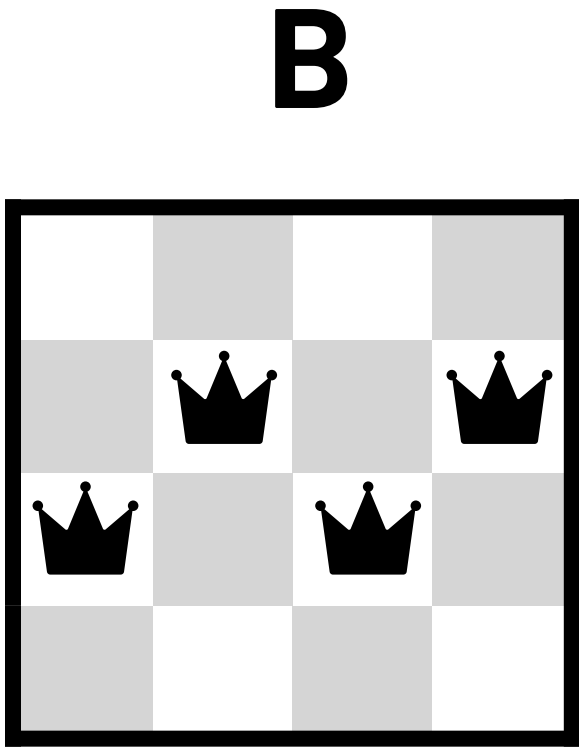
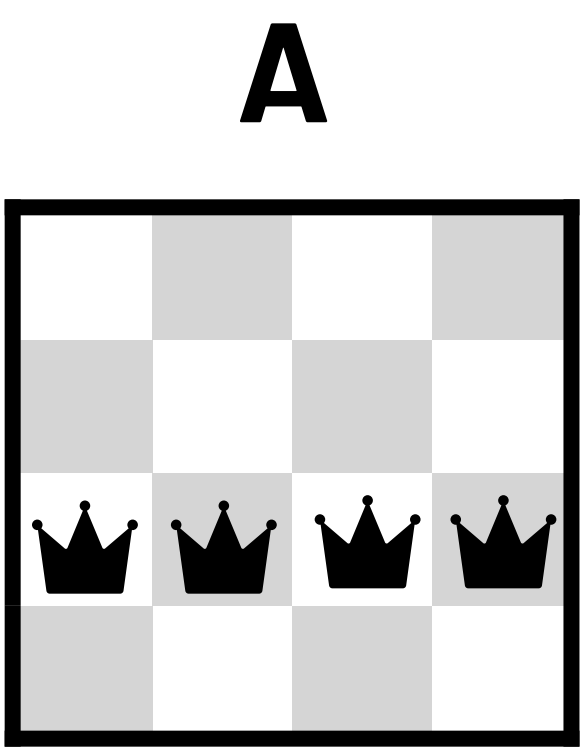
- ▶ **Roleta**
- ▶ **Torneio**
- ▶ **Ranking**



# Seleção por roleta

Na seleção por **roleta**, os pais são selecionados aleatoriamente com probabilidade proporcional ao valor de adaptação.

	Roleta	Selecionado
1		C
2		D
...	...	...
m		A



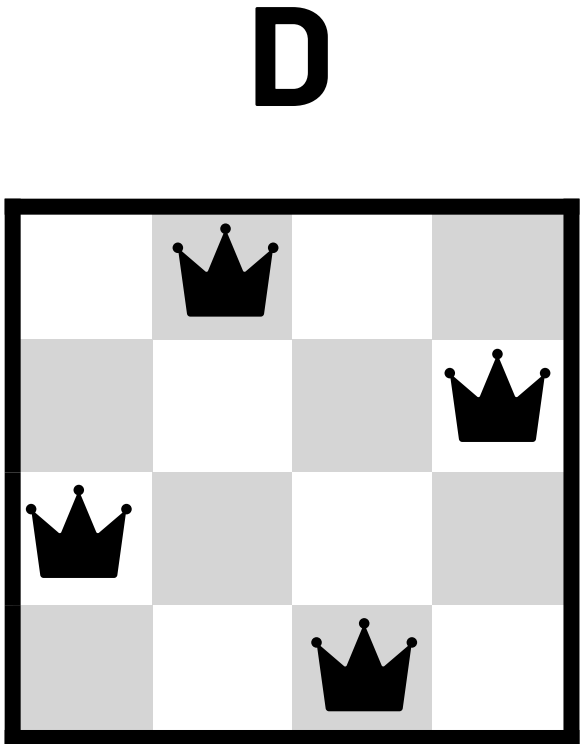
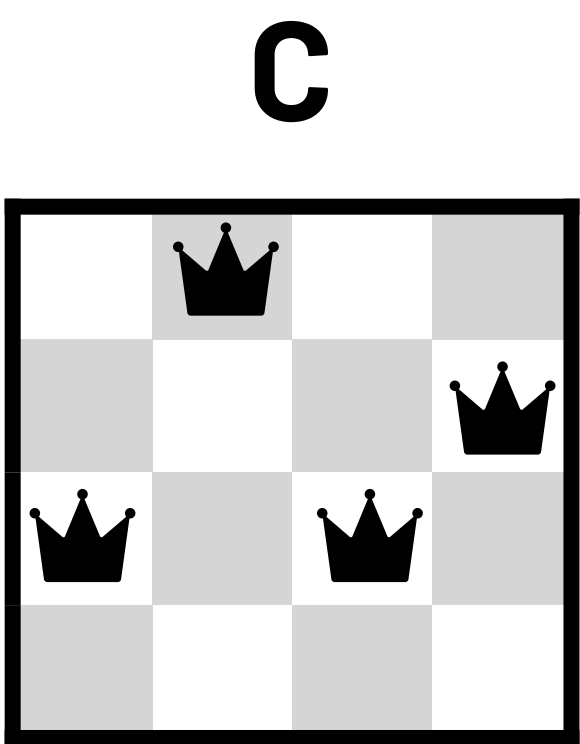
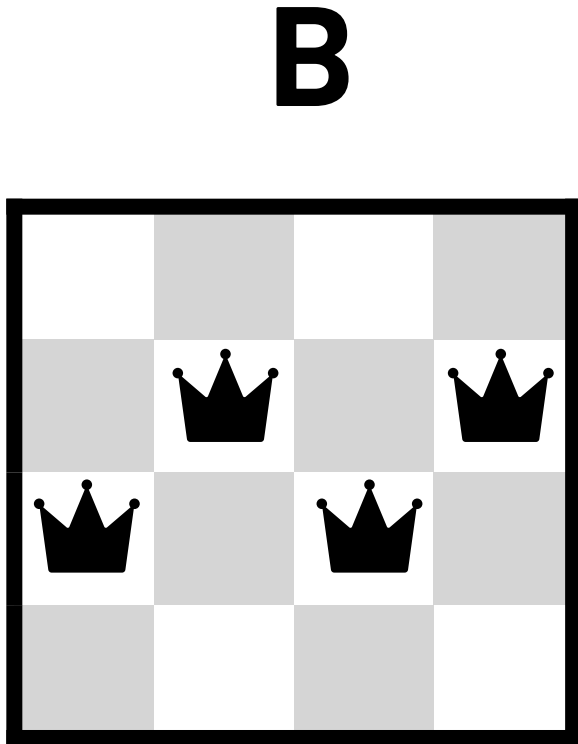
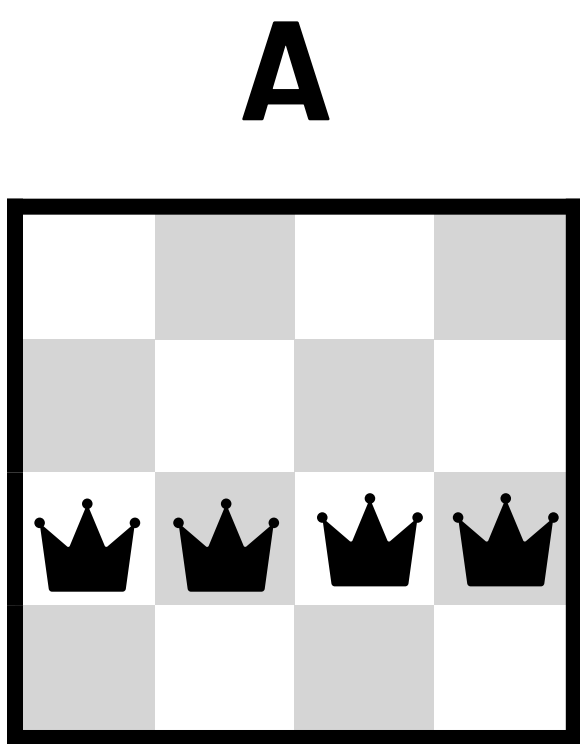
	<b>Conflitos (h)</b>	6	5	2	0
	<b>Adaptação (f)</b>	0,14	0,16	0,33	1
$f = 1/(1 + h)$	<b>Probabilidade</b>	9%	10%	20%	61%

Como o n-rainhas é um problema de minimização, a função de adaptação é inversamente proporcional ao número de conflitos

# Seleção por torneio

Na seleção por **torneio**, os pais são selecionados via torneios compostos por  $k$  indivíduos aleatórios. O indivíduo com maior adaptação é escolhido como vencedor de cada torneio.

Torneios (k=2) Selecionado			
1	B	×	C
2	A	×	D
...	...		...
m	A	×	C

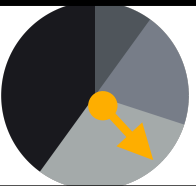

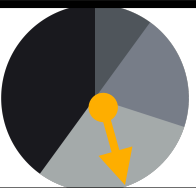


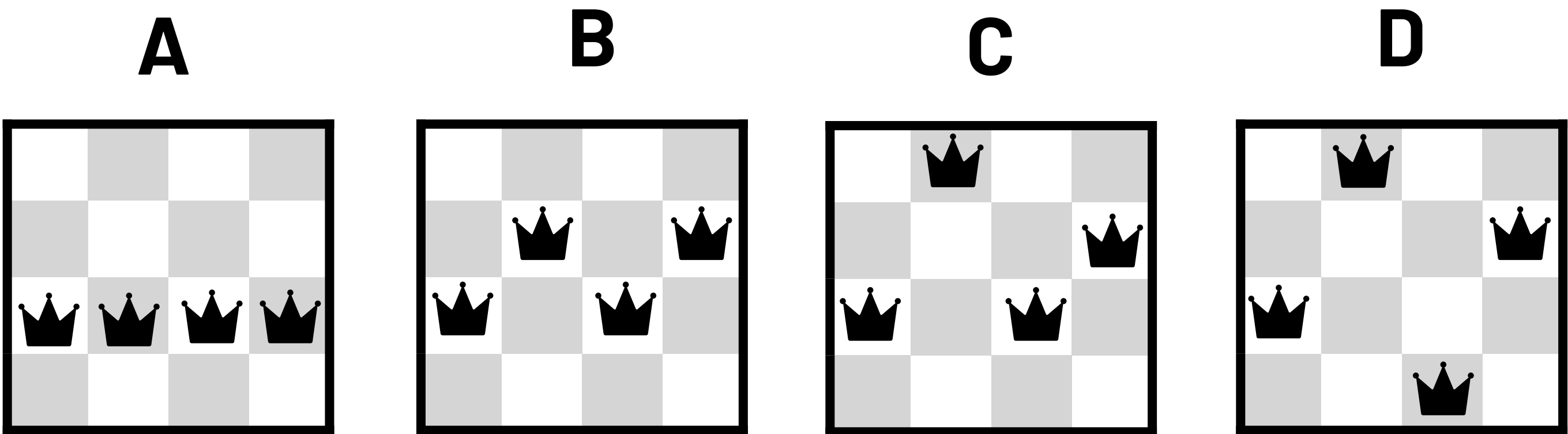
Conflitos (h)	6	5	2	0
Adaptação (f)	0,14	0,16	0,33	1



# Seleção por ranking

Na seleção por **ranking**, os pais são ordenados por valor de adaptação e selecionados aleatoriamente com probabilidade proporcional a sua posição no ranking.

	Ranking	Vencedor
1		C
2		D
...	...	...
m		C



Conflitos (h)	6	5	2	0
Adaptação (f)	0,14	0,16	0,33	1
Ranking	1	2	3	4
Probabilidade	10%	20%	30%	40%

# Seleção elitista



- ▶ Os métodos de seleção probabilística que vimos não garantem que o melhor indivíduo da população sobreviverá até à próxima geração.
- ▶ Uma solução para este problema é chamada de **elitismo** e consiste em manter sempre o melhor (ou os k melhores) indivíduo na nova geração.

# Cruzamento

A etapa de **cruzamento** produz uma nova geração de indivíduos escolhendo pares de pais do grupo de acasalamento e combinando seu genótipo. Existem três métodos de cruzamento mais comuns:

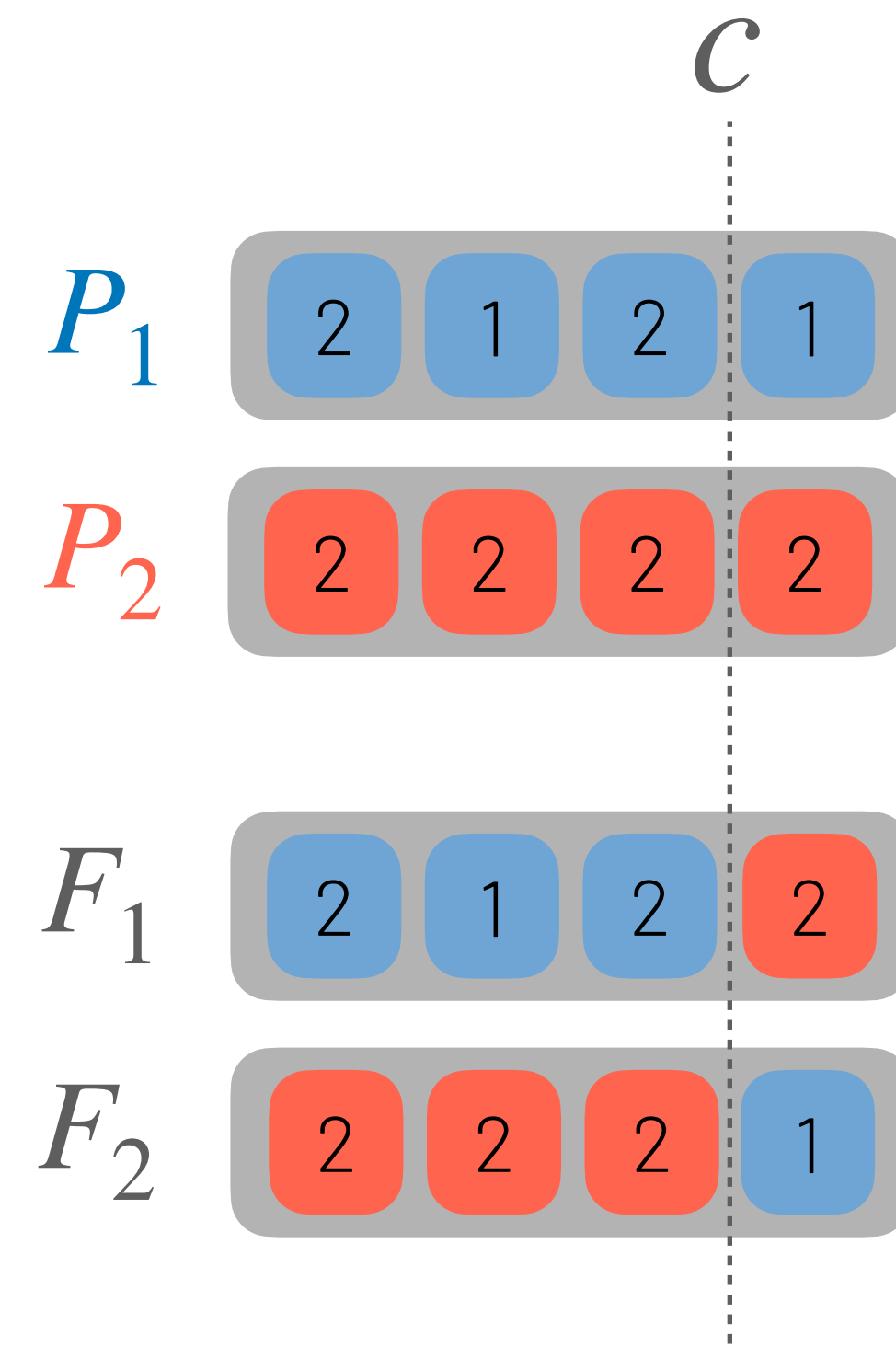
- ▶ **Cruzamento com um ponto de corte**
- ▶ **Cruzamento com dois pontos ou  $k$ -pontos de corte**
- ▶ **Cruzamento uniforme**



# Cruzamento com um ponto de corte

Dados dois indivíduos pais  $P_1$  e  $P_2$  com genótipo de tamanho  $d$ , o **cruzamento com um ponto de corte** produz dois novos filhos  $F_1$  e  $F_2$  da seguinte maneira:

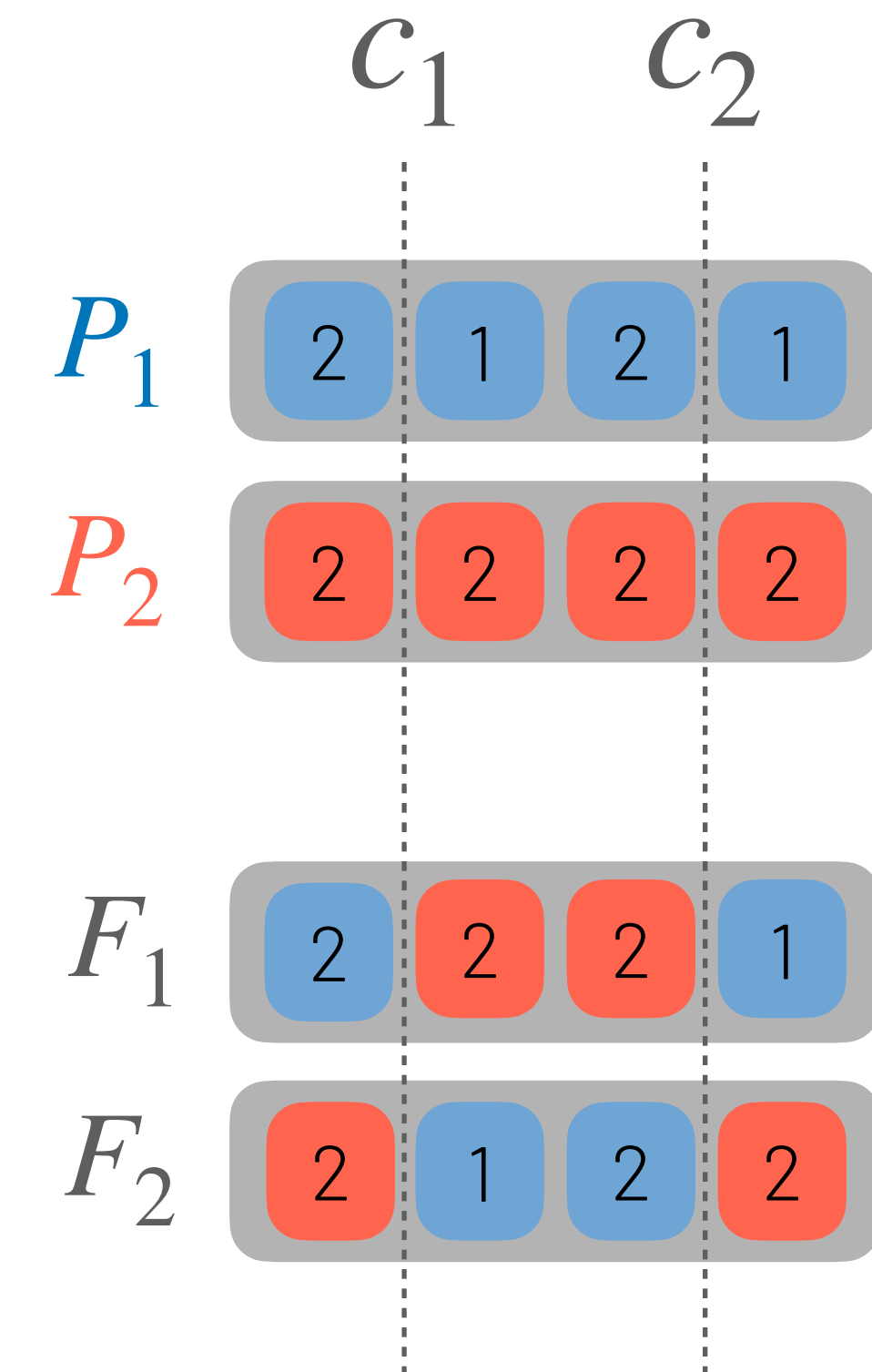
1. Selecione aleatoriamente um **ponto de corte**  $0 \leq c < d$
2. Produza  $F_1$  concatenando  $P_1[0..c]$  com  $P_2[c..d]$
3. Produza  $F_2$  concatenando  $P_1[c..d]$  com  $P_2[0..c]$



# Cruzamento com $k$ -pontos de corte

Dados dois indivíduos pais  $P_1$  e  $P_2$  com genótipo de tamanho  $d$ , o **cruzamento com  $k$ -pontos de corte** produz dois filhos  $F_1$  e  $F_2$  da seguinte maneira:

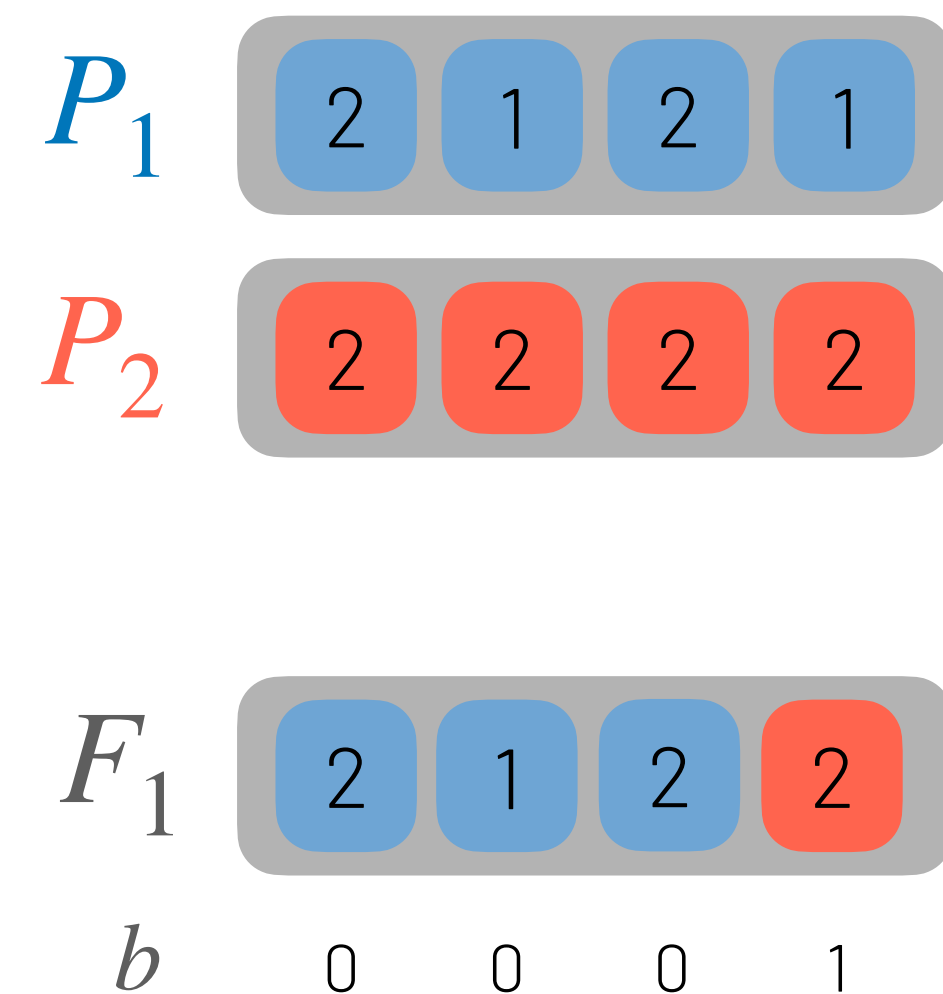
1. Selecione aleatoriamente  $k$  **pontos de corte**  $c_1, \dots, c_k$ ,  $0 \leq c_i < d$
2.  $c = \{c_0, c_1, c_2, \dots, c_k, c_{k+1}\}$ , onde  $c_0 = 0$  e  $c_{k+1} = d$
3.  $F_1 = [], F_2 = []$
4. **para**  $i = 0$  até  $k$ :
5.   **se**  $i$  é par:
6.     Concatene  $P_1[c_i \dots c_{i+1}]$  em  $F_1$  e  $P_2[c_i \dots c_{i+1}]$  em  $F_2$
7.   **senão**:
8.     Concatene  $P_2[c_i \dots c_{i+1}]$  em  $F_1$  e  $P_1[c_i \dots c_{i+1}]$  em  $F_2$



# Cruzamento uniforme

Dados dois indivíduos pais  $P_1$  e  $P_2$  com genótipo de tamanho  $d$ , o **cruzamento uniforme** produz **um** filho  $F_1$  da seguinte maneira.

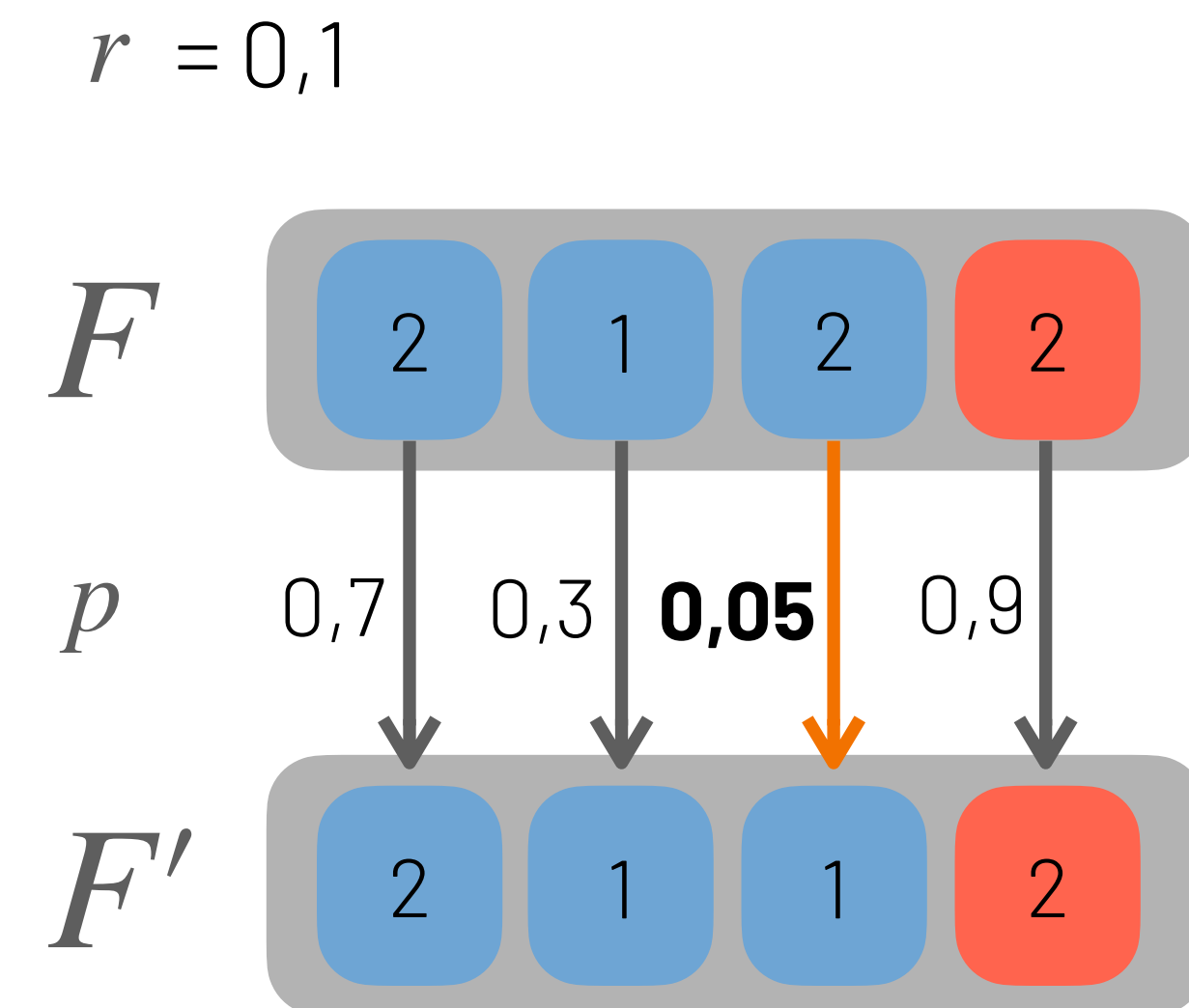
1.  $F_1 = [], F_2 = []$
2. **para**  $i = 0$  até  $d$ :
3. Gere um número bit  $b$  aleatoriamente
4. **se**  $b == 0$ :
5. Concatene  $P_1[i]$  em  $F_1$
6. **senão**:
7. Concatene  $P_2[i]$  em  $F_1$



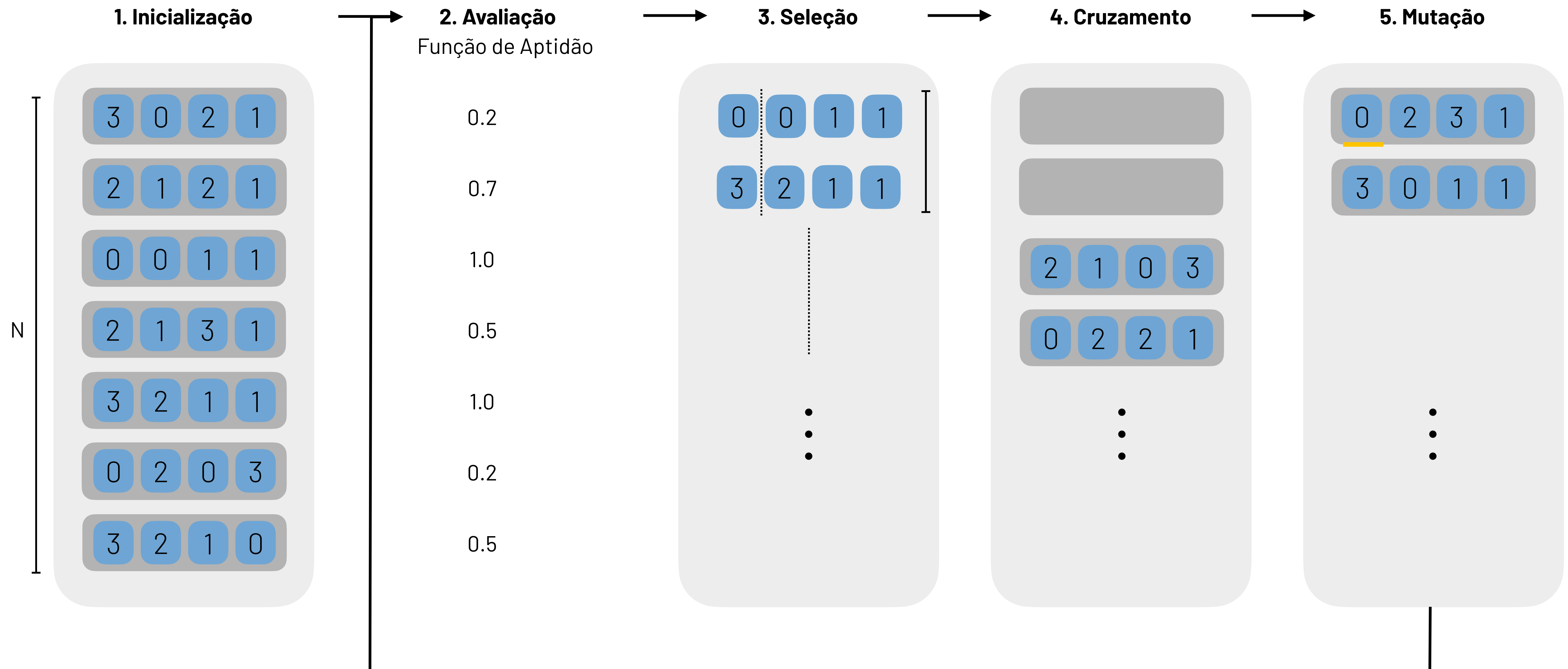
# Mutação

- ▶ A etapa de **mutação** introduz pequenos ruídos nos filhos gerados por cruzamento, visando evitar que o algoritmo fique preso em ótimos locais.
- ▶ Dado um indivíduo filho  $F$  com genótipo de tamanho  $d$ , esse processo pode ser feito da seguinte maneira:

1. Defina uma taxa de mutação  $r$  (geralmente 1-5%):
2. **para**  $i = 0$  até  $d$ :
3. Gere um número  $p$  entre 0 e 1
4. **se**  $p < r$ :
5.  $F[i]$  recebe um valor de gene aleatório



# Algoritmos Genéticos



# Aplicações de algoritmos genéticos

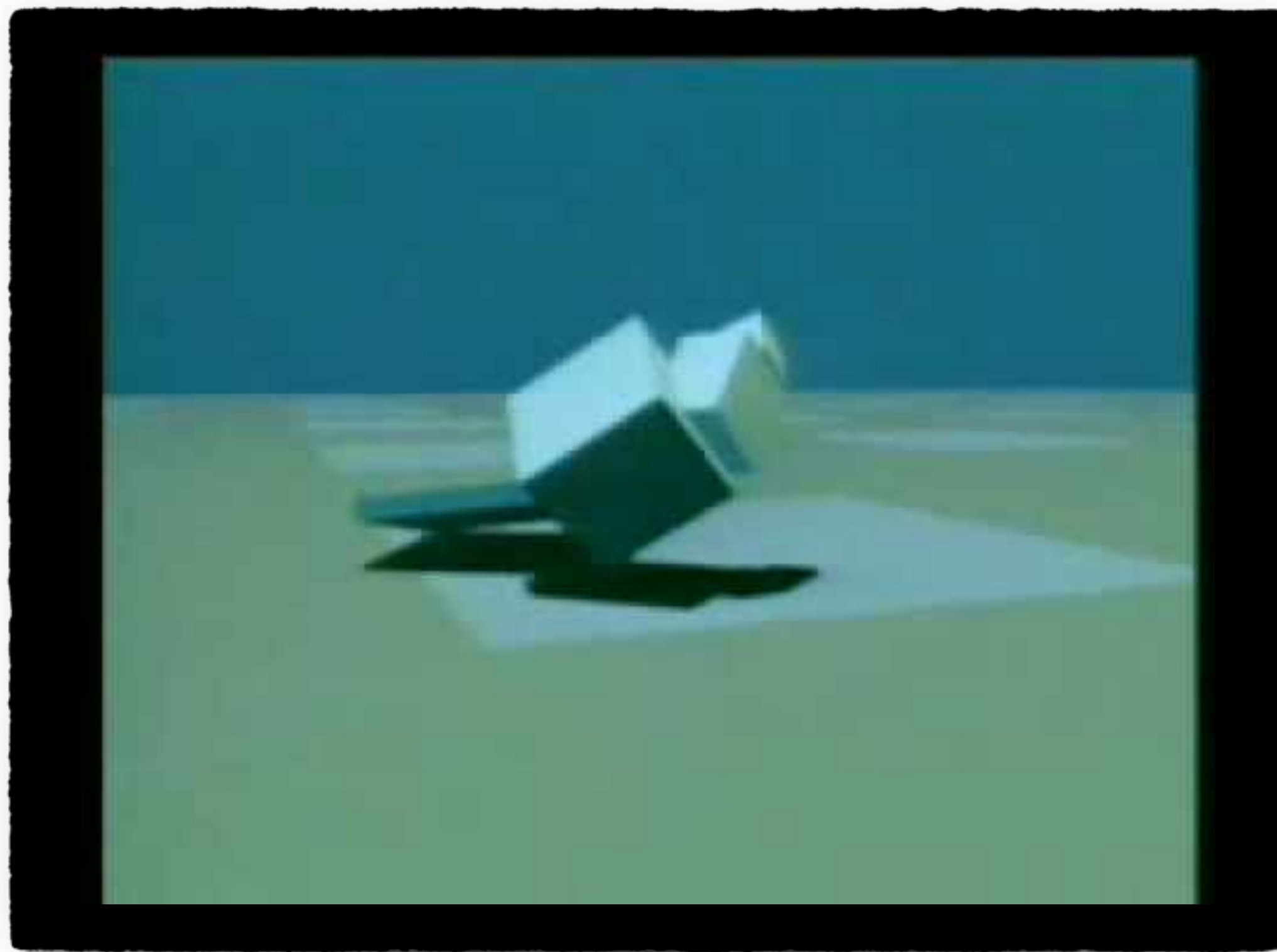


## Restilização de imagens

- ▶ Imagens representadas por polígonos
- ▶ Função de aptidão direta que calcula a distância pixel-a-pixel entre o fenótipo do indivíduo e a imagem objetivo
- ▶ Exemplo:  
<https://chriscummins.cc/s/genetics/>



# Aplicações de algoritmos genéticos



## Geração de criaturas

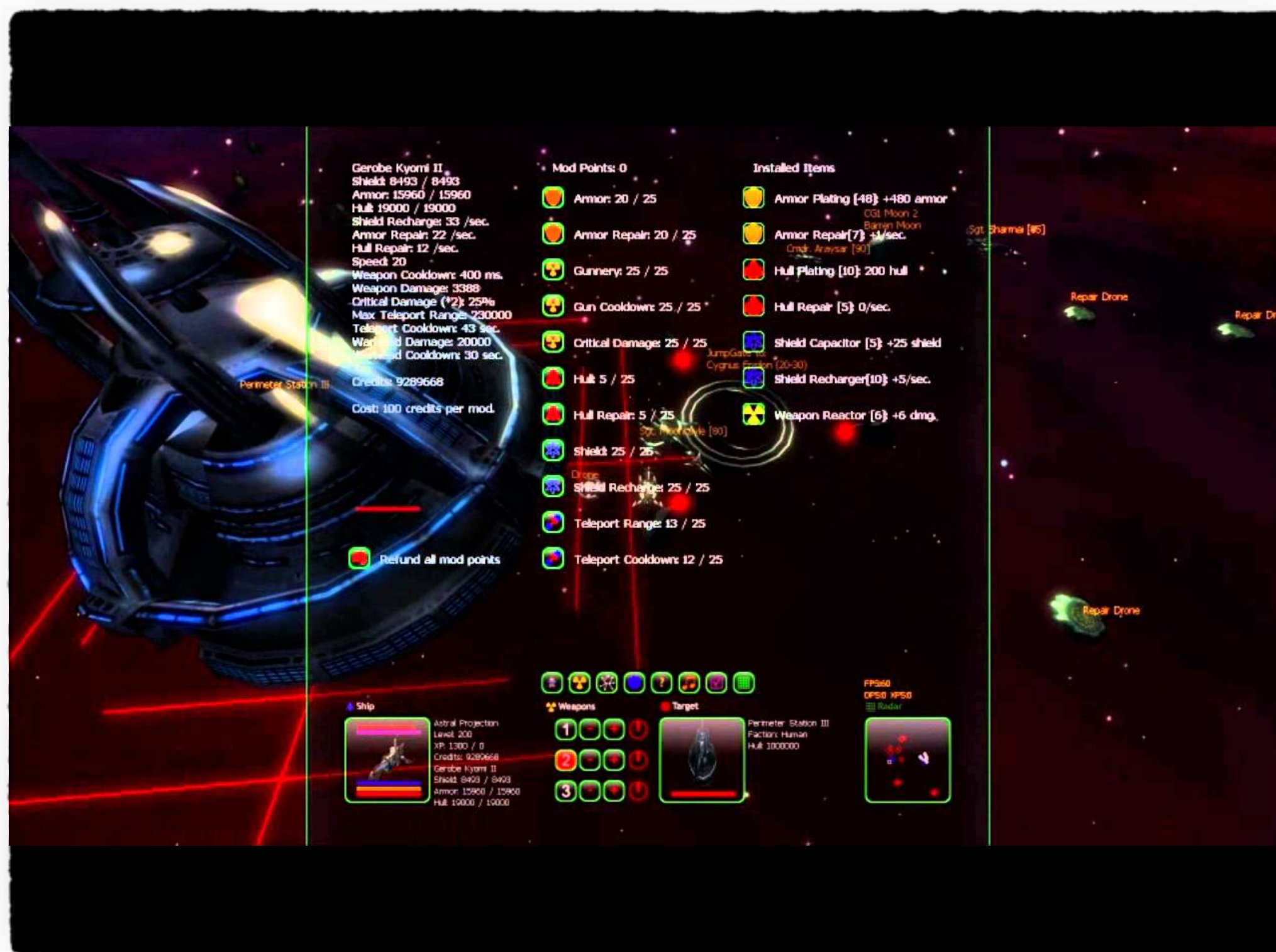
- ▶ Criaturas representadas por sólidos geométricos
- ▶ Função de aptidão baseada em simulação medindo o quanto a criatura se moveu
- ▶ Exemplo:  
<https://keiwan.itch.io/evolution>

# Aplicações de algoritmos genéticos

## Galactic Arms Race (jogo MMO de batalha)

- ▶ Evolução de armas representadas como sistemas de partículas
- ▶ Função de aptidão interativa, onde a qualidade das armas é medida pelo tempo que os usuários utilizam

[https://store.steampowered.com/app/249610/Galactic\\_Arms\\_Race/](https://store.steampowered.com/app/249610/Galactic_Arms_Race/)





# Próxima aula

## **A6:** Busca competitiva I

Teoria dos jogos; jogos como problemas de busca; algoritmo minimax; poda alpha-beta e função de avaliação