

ALGORITMOS GENÉTICOS

Metaheurísticas de Buscas

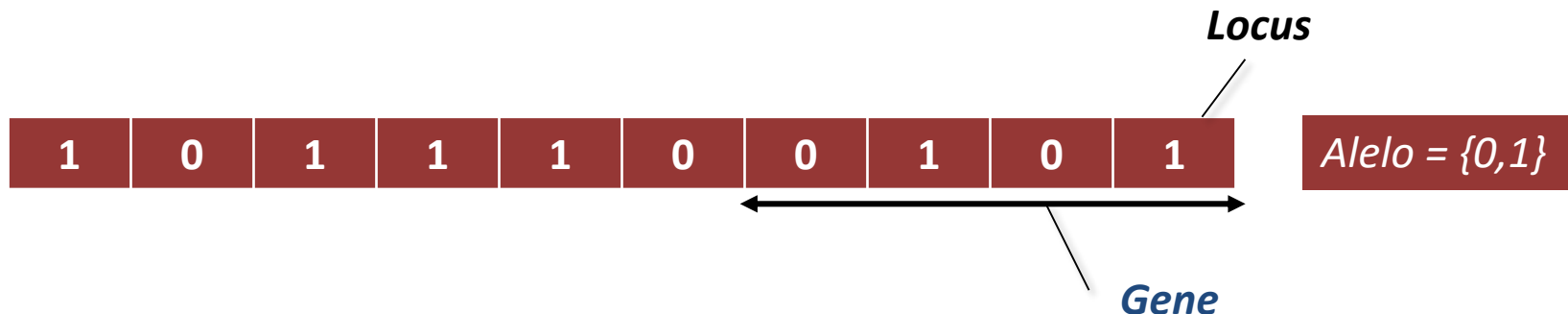
ALGORITMOS GENÉTICOS (AG)

Popularizados por **John Holland** podem ser considerados os primeiros modelos algorítmicos desenvolvidos para inspirados nos sistemas genéticos.

Publicou em 1975 o livro seminal de AG:
Adaptation in Natural and Artificial Systems

ALGORITMOS GENÉTICOS (AG)

- **CROMOSSOMO (ou indivíduo)**: formado por genes; normalmente um cromossomo representa uma solução completa a um problema.
- **GENE**: representa uma **faceta/característica** independente das demais.
- **ALELO**: é o **valor** armazenado em um locus.



ALGORITMOS GENÉTICOS (AG)

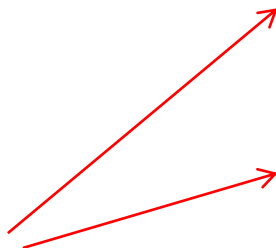
- **POPULAÇÃO**: conjunto de cromossomos ou de indivíduos.
- **SELEÇÃO**: Sobrevivência do melhor; manter os indivíduos de maior adequabilidade (fitness) da população.
- **REPRODUÇÃO**: realizada pelos operadores abaixo
Crossover: cruzamento entre pares de cromossomos
Mutação: modificação de genes de um cromossomo

AG no contexto de Busca em IA

CARACTERÍSTICAS

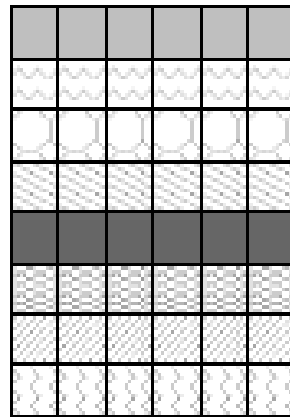
- **Busca informada:** utiliza função de *fitness* que guia o algoritmo
- **Busca local:** não guarda a informação do caminho para chegar à solução
- **Estados sucessores** gerados pela combinação de dois estados pais
- **Função de adequação ou fitness:** $f(n) = h(n)$
 $f(\text{indivíduo}) = \text{função de adequação ou fitness (indivíduo)}$
- **Útil quando** o espaço de estados é muito grande ou muito complexo para tratamento analítico

ESQUEMA GERAL DE EVOLUÇÃO DE UM ALGORITMO GENÉTICO

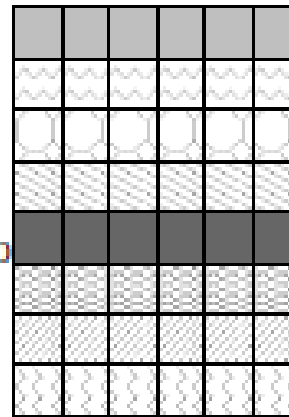


Esquema Geral de um Algoritmo Genético

População Inicial N=8



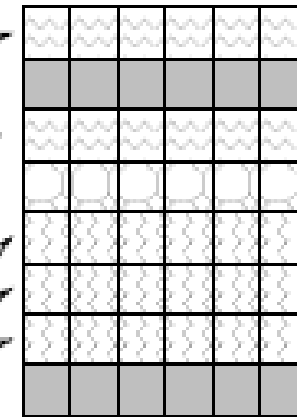
Avalia
Desempenho



Fitness MaxSorteiosDaRoleta=N=8

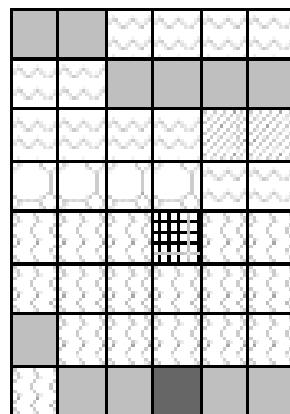
7
9
5
1
2
1
3
10

Seleção

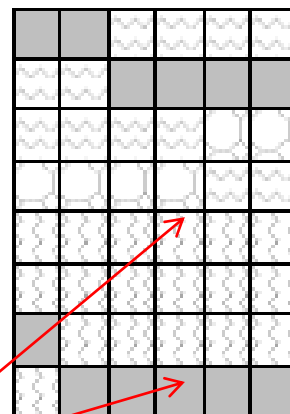


Atualiza
Geração

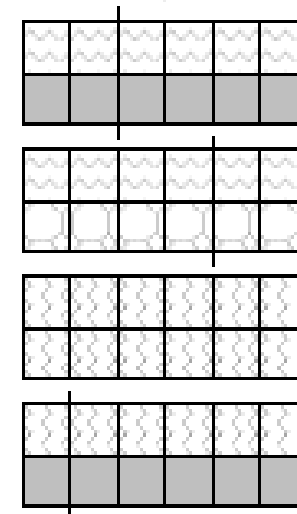
Nova População



Mutação



Crossover



Escolha de
Pares e Pontos
de Crossover

$r1 < PrCr$

$r2 < PrCr$

$r3 > PrCr$

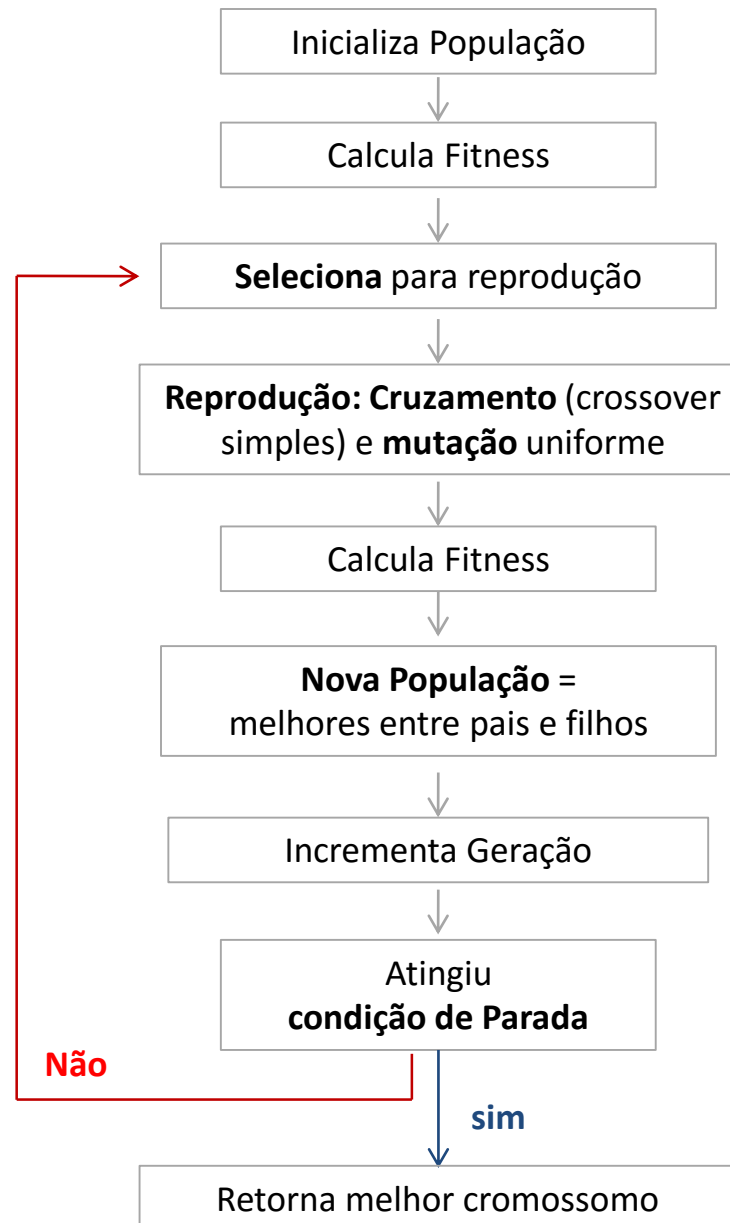
$r4 < PrCr$

$r5 < PrMut$

aleatório [0,1]

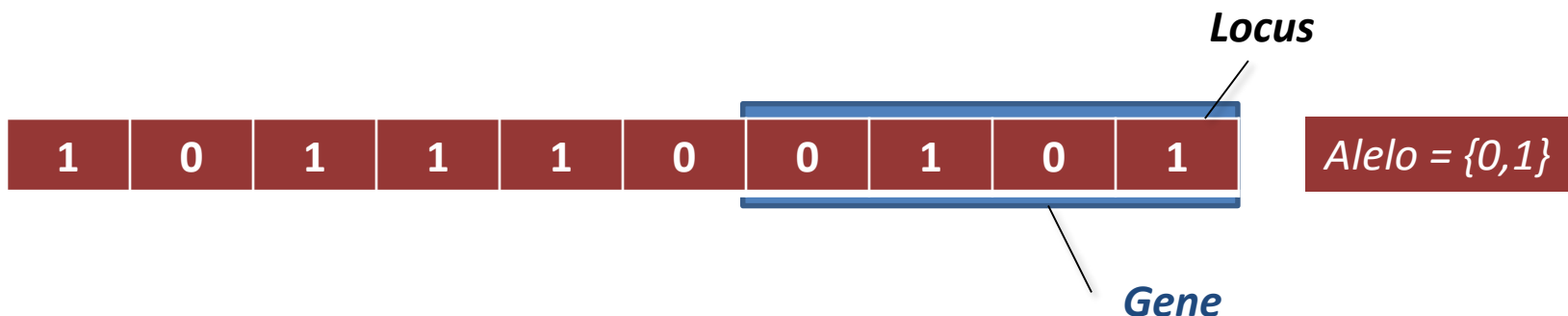
AG Canônico

- Codificação Binária
- **Seleção**
 - Reprodução (método da roleta)
- **Reprodução**
 - Crossover simples (pCROSS)
 - Mutação Uniforme (pMUT)
- **Sobrevivência**
 - Melhores (entre pais e filhos) irão compor a nova população
- **Condição de parada**
 - Geração = MaxGer ou
 - Fitness = máximo atingido ou
 - Estagnação do melhor fitness



AG Canônico: Codificação binária

Cromossomo no AG: cadeia de bits de tamanho L



Cada posição (locus) no cromossomo assume um dos dois possíveis alelos, 0 ou 1.

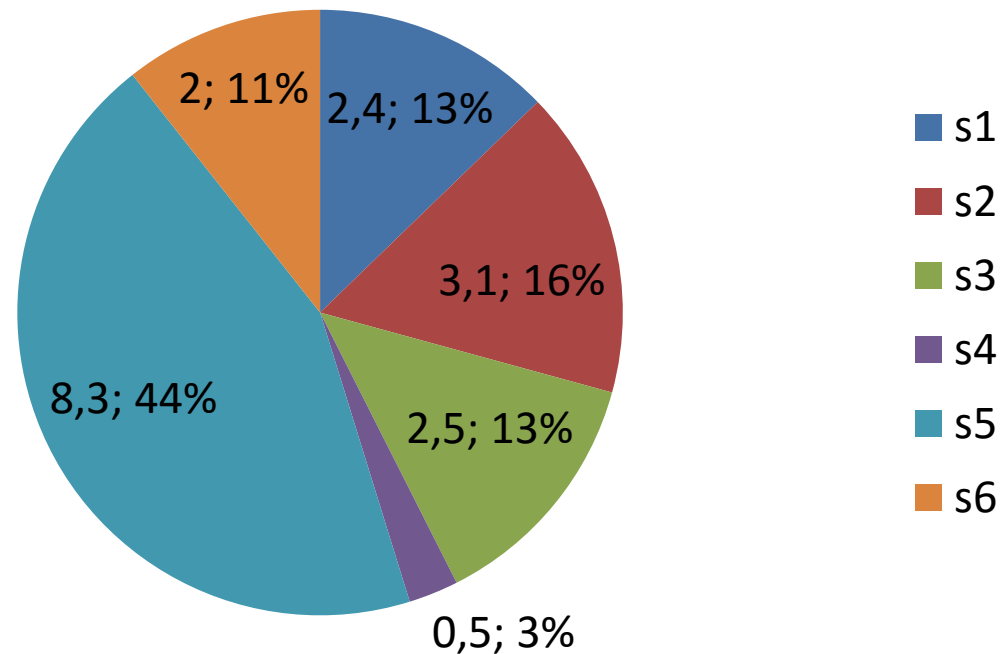
Exemplo: problema da mochila; cada gene indica se um determinado item está ou não na mochila

AG Canônico: seleção por roleta

Método de seleção de cromossomos/indivíduos que seleciona K indivíduos , sendo que os de maior probabilidade são os que apresentam maior *fitness*.

Roleta

Indivíduo	Fitness	Norm.
S1	2,4	13%
S2	3,1	16%
S3	2,5	13%
S4	0,5	3%
S5	8,3	44%
S6	2,0	11%



AG Canônico: Seleção por roleta

Calcula $p(s_i)$ para $i=1,\dots,N$:
$$p(s_i) = \frac{fitness(s_i)}{\sum_{k=1}^N fitness(s_k)}$$

Algoritmo de seleção por roleta: sorteia um indivíduo por chamada

```
i=1;  
soma = p(si)  
Sorteia r ∈ [0, 1]  
enquanto soma < r  
    i=(i+1)  
    soma = soma + p(si)  
fim enquanto  
Retorna si
```

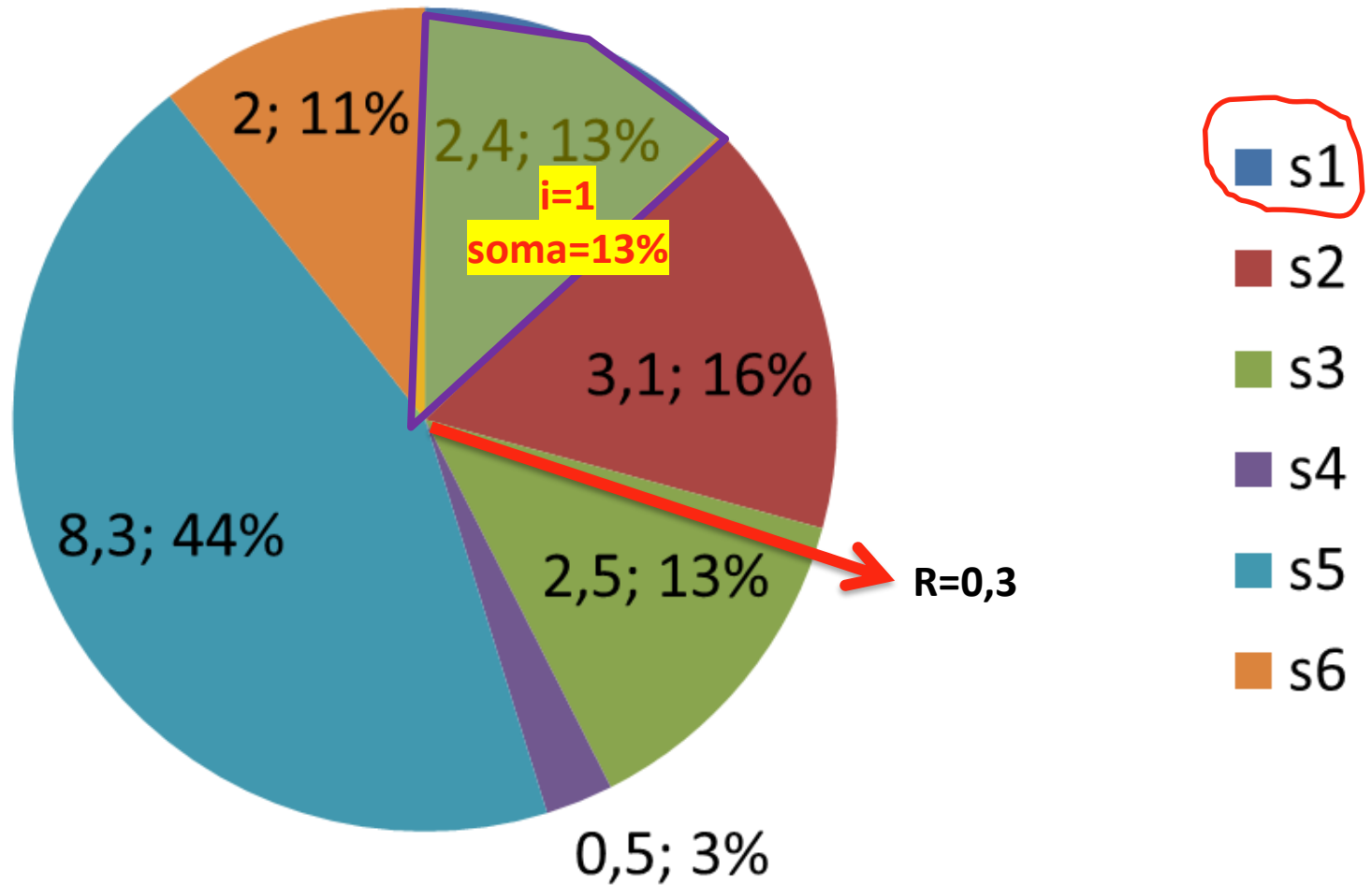
INICIALIZAÇÃO

soma := $p(s1) = 13\%$

$r = 30\%$

$13\% < 30\%$

Roleta



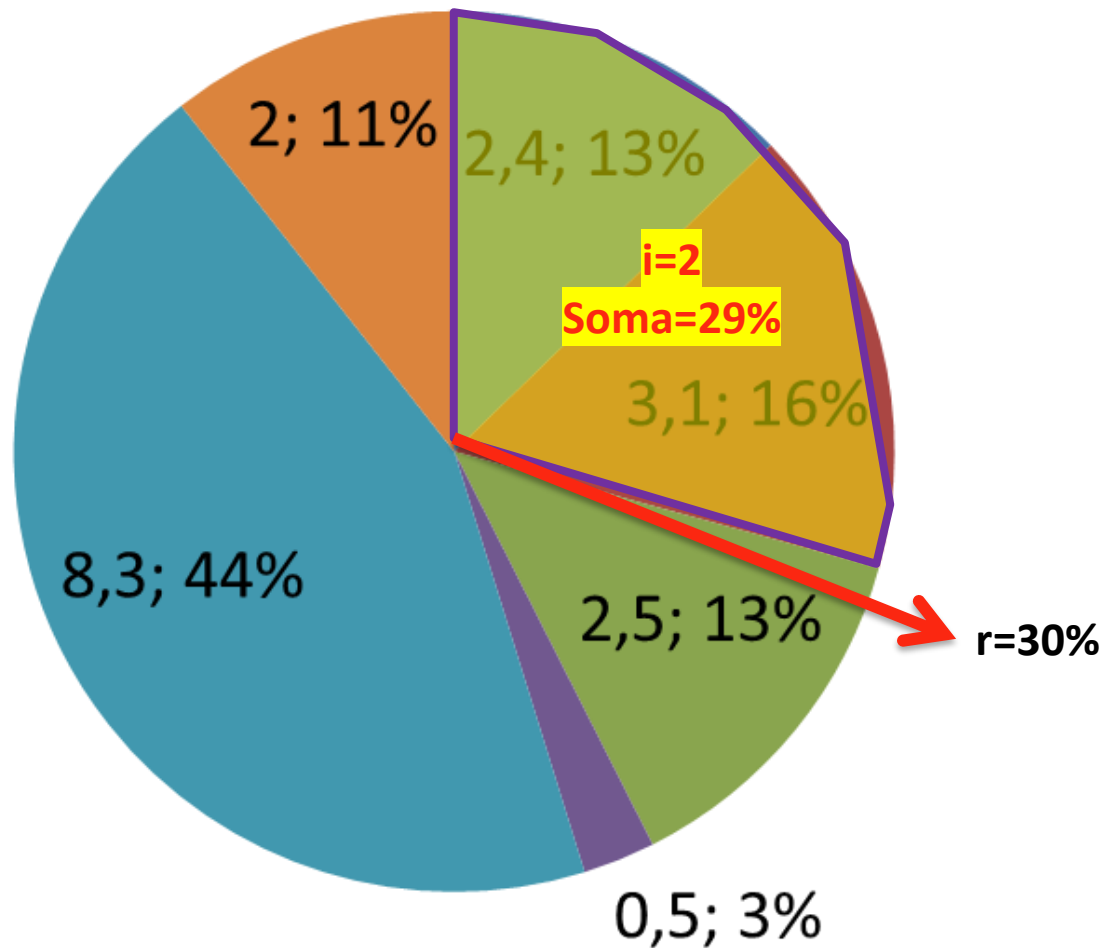
SORTEIO

MaxSorteiosRoleta=4

Soma=29%

29% < 30%

Roleta



s1

s2

s3

s4

s5

s6

SORTEIO

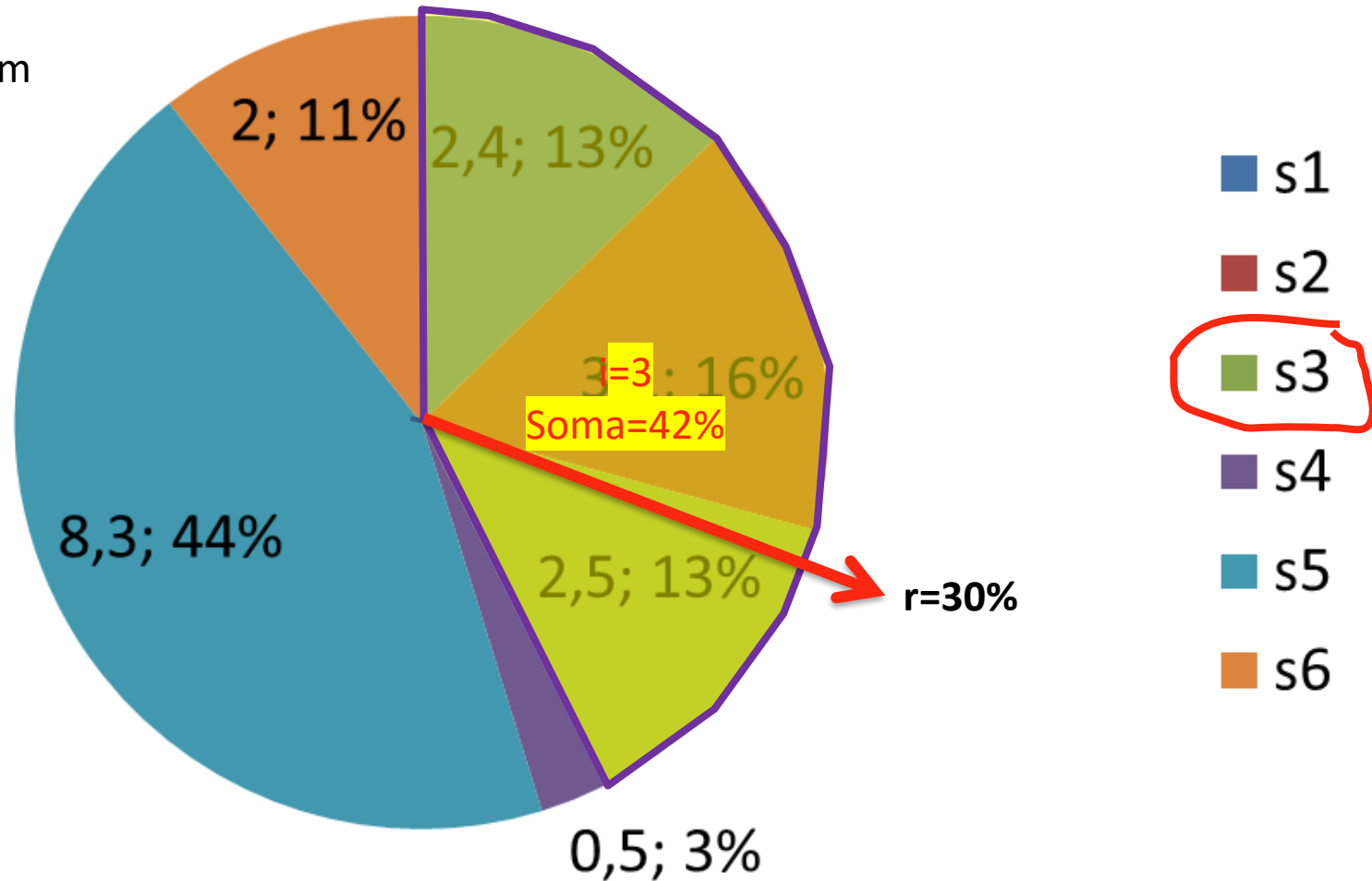
MaxSorteiosRoleta=4

Soma=42%

42% > 30% => fim

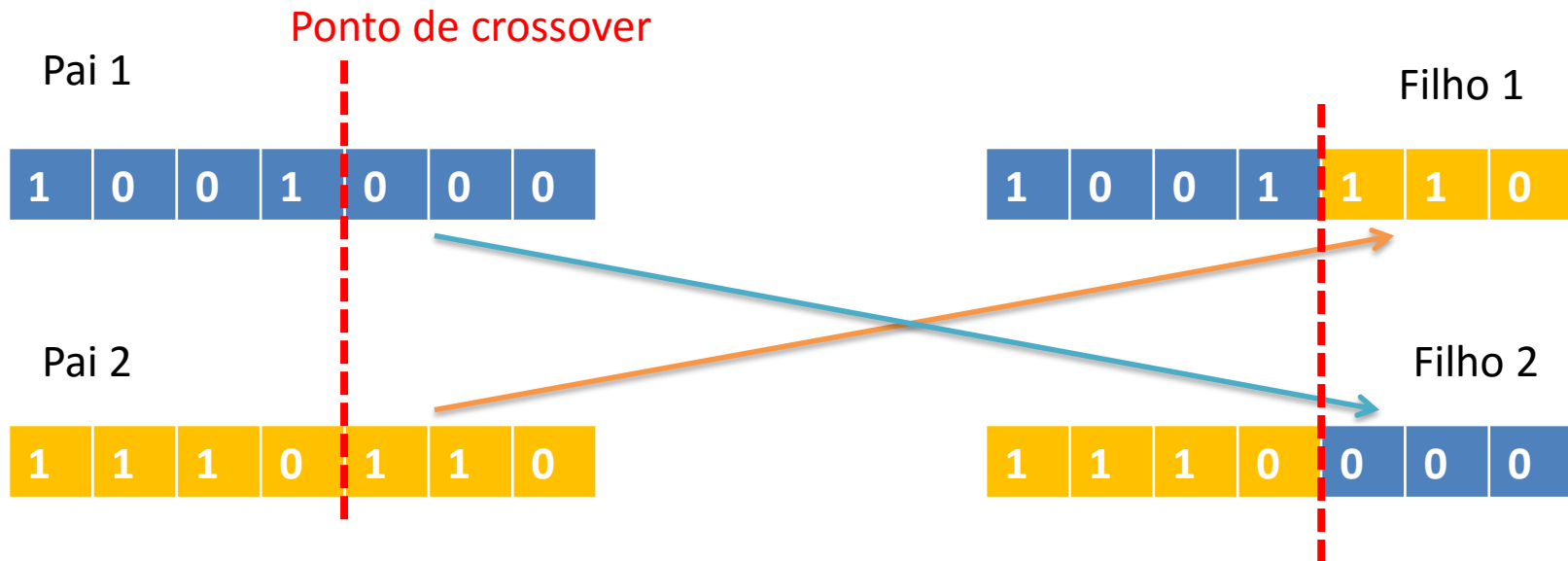
Sorteado=[s3]

Roleta

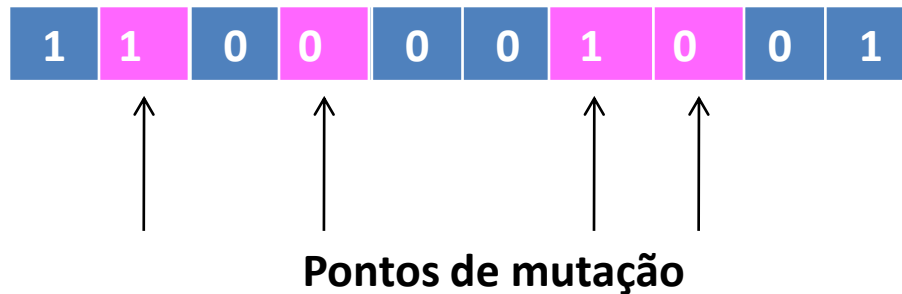


AG Canônico: Reprodução e crossover

Crossover Simples (1 ponto): posição de cruzamento escolhida aleatoriamente.



AG Canônico: Reprodução e Mutação

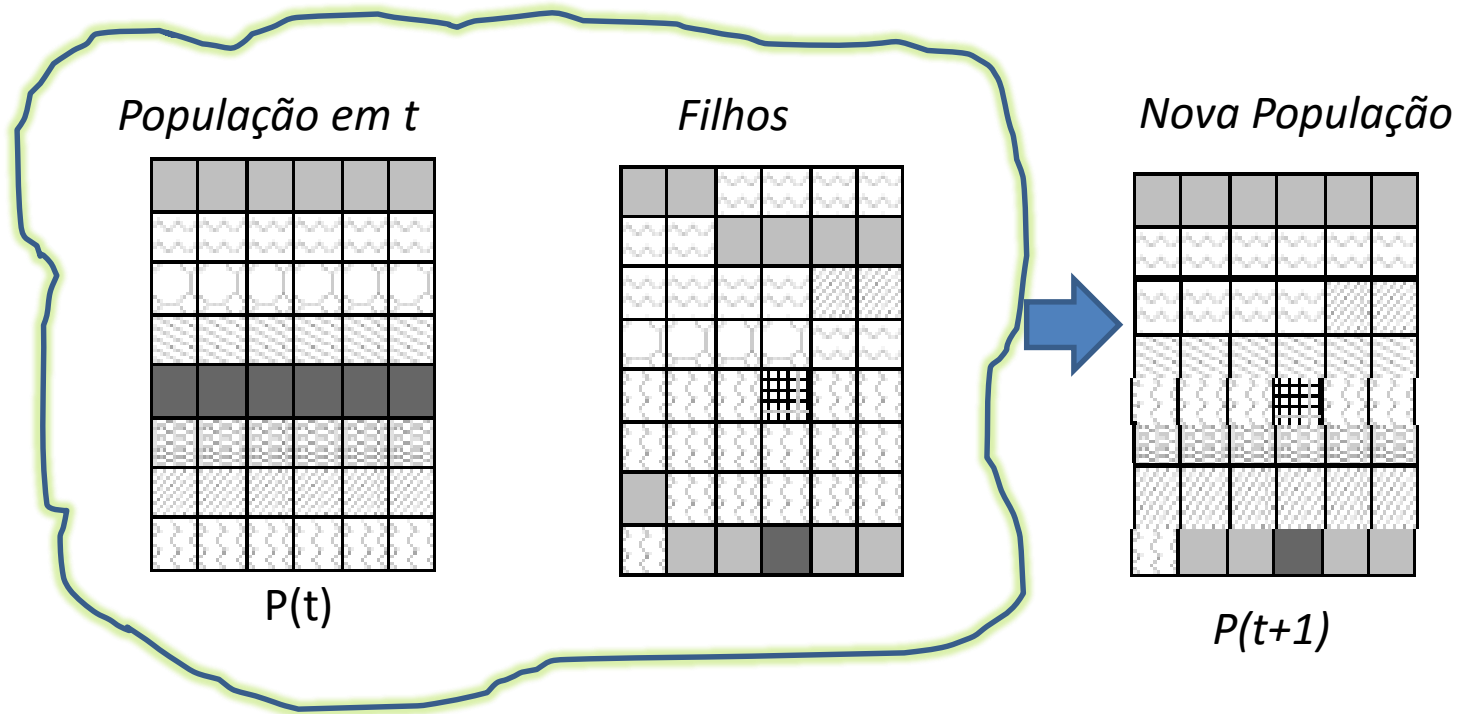


Posições são **escolhidas aleatoriamente**.

Mutação realizada com **probabilidade baixa**
(tipicamente 0.05)

AG Canônico: sobrevivência

Sobrevivência dos Melhores: selecionar entre **pais e descendentes (filhos)** para compor a nova população



$$P(t+1) = \text{Melhores}(P(t) \cup \text{Filhos})$$

AG Canônico: pseudo-código

C := { c_1, \dots, c_n } população inicial de tamanho **N**
R tamanho da descendência ou reprodução (normalmente $K = N$)
pCROSS: probabilidade de fazer crossover (valor típico 0.8)
pMUT: Probabilidade de fazer mutação nos alelos (valor típico 0.05)

AGCanônico(C, N, R, pCROSS, pMUT) {

Para todo c_i de C, calcular **fitness**(c_i);

geracao:=0;

do {

// D = Descendentes = nova geração, calculada a partir de C

D := selecionar **R** cromossomos de C pelo **método da roleta**;

D' := cruzamento(D); *// geração de dois filhos por par (d_1, d_2), (d_3, d_4), ..., (d_{k-1}, d_k)
de D fazendo crossover com probabilidade **pCROSS***

D'' := para todo cromossomo d_i de D', para cada alelo a_j de d_i , **mutar** a_j
com probabilidade **pMUT**;

Para todo d_i de D'', calcular **fitness**(d_i);

// Selecionar melhores entre pais e filhos

C := selecionar **n** melhores cromossomos de $C \cup D''$;

geracao++;

} while (geracao < MAX_GERACOES and !objetivo-alcançado and
!melhor-fitness estagnado);

retornar c_i de C com melhor fitness;

}

AG Canônico: resumo

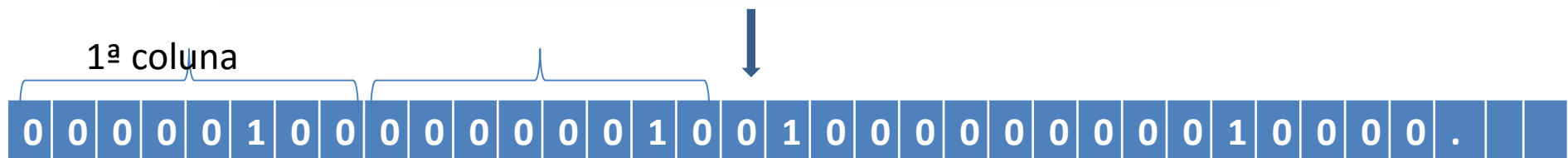
- Representação do indivíduo: codificação binária
- Seleção
 - Roleta
- Reprodução
 - Crossover simples com seleção aleatória de pares (pais)
 - Mutação Uniforme
- Sobrevivência
 - Melhores entre pais e filhos para compor a nova população

EXEMPLOS

PROBLEMA DAS 8 RAINHAS

Solução por Codificação binária (matriz binária) -> vetor de bits

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



PROBLEMA DAS 8 RAINHAS

Solução por codificação binária (obtida por conversão inteiro-> binário)

8							
7		R					
6							
5			R				
4					R		
3	R						
2		R		R			
1						R	R



011	010	110	101	010	100	001	001
3	2	7	5	2	4	1	1

PROBLEMA DAS 8 RAINHAS

Solução por Codificação inteira

		R					
			R				
					R		
R							
	R			R			
						R	R



3	2	7	5	2	4	1	1
---	---	---	---	---	---	---	---

Problema da Mochila

Problema da mochila: Um viajante deve levar consigo, apenas uma mochila. Essa mochila possui uma capacidade limitada e deve ser carregada apenas com objetos que serão úteis durante a viagem. Cada objeto é único e possui um peso e um determinado valor.

Que objetos devem ser levados pelo viajante de forma a maximizar o valor da mochila?

Problema da mochila

$$\begin{aligned} &\text{maximizar } f(x) = \sum_{i=1}^n x_i v_i \\ &\text{sujeito a } \sum_{i=1}^n x_i w_i \leq C, x_i \in \{0, 1\} \end{aligned} \quad (1)$$

onde v_i é o valor i-ésimo objeto, w_i o seu peso, x_i indica se o objeto aparece ou não na mochila e C define a capacidade da mochila (em termos de peso).

Obj(i=1,...n)	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Peso(kg)	3	8	12	2	8	4	4	5	1	1	8	6	4	3
Valor	1	3	1	8	9	3	2	8	5	1	1	6	3	2
Obj(i=1,...n)	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Peso(kg)	3	5	7	3	5	7	4	3	7	2	3	5	4	3
Valor	5	2	3	8	9	3	2	4	5	4	3	1	3	2

Capacidade da mochila=28Kg

Problema da mochila

Solução por Codificação binária (matriz binária) -> vetor de bits

- item presente: 1
- item ausente: 0

ITENS

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

**AG EM PROBLEMAS DE
PERMUTAÇÃO**

Problemas de permutação

- **Solução** = determina a ordem de uma sequência de eventos
- **Exemplos**
 - **caixeiro viajante**: *determinar a ordem de visita das cidades minimizando a distância percorrida (problema de otimização de rotas)*
 - **scheduling**: *ordem de operações para minimizar tempo de produção de peças)*
- *Que codificação utilizar?*
- *Quais operadores utilizar?*

Problemas de Permutação

Normalmente utiliza-se

Codificação da ordem: gene indica a ordem de visitação de cada uma das cidades de A a I
gene=ordem de visita alelo=[1, número de cidades]

Caixeiro viajante

cidade	A	B	C	D	E	F	G	H	I	
ordem visita	5	2	9	4	7	6	1	3	8	= cromossomo

Codificação posicional: gene indica a cidade que será visitada em 1º, 2º,

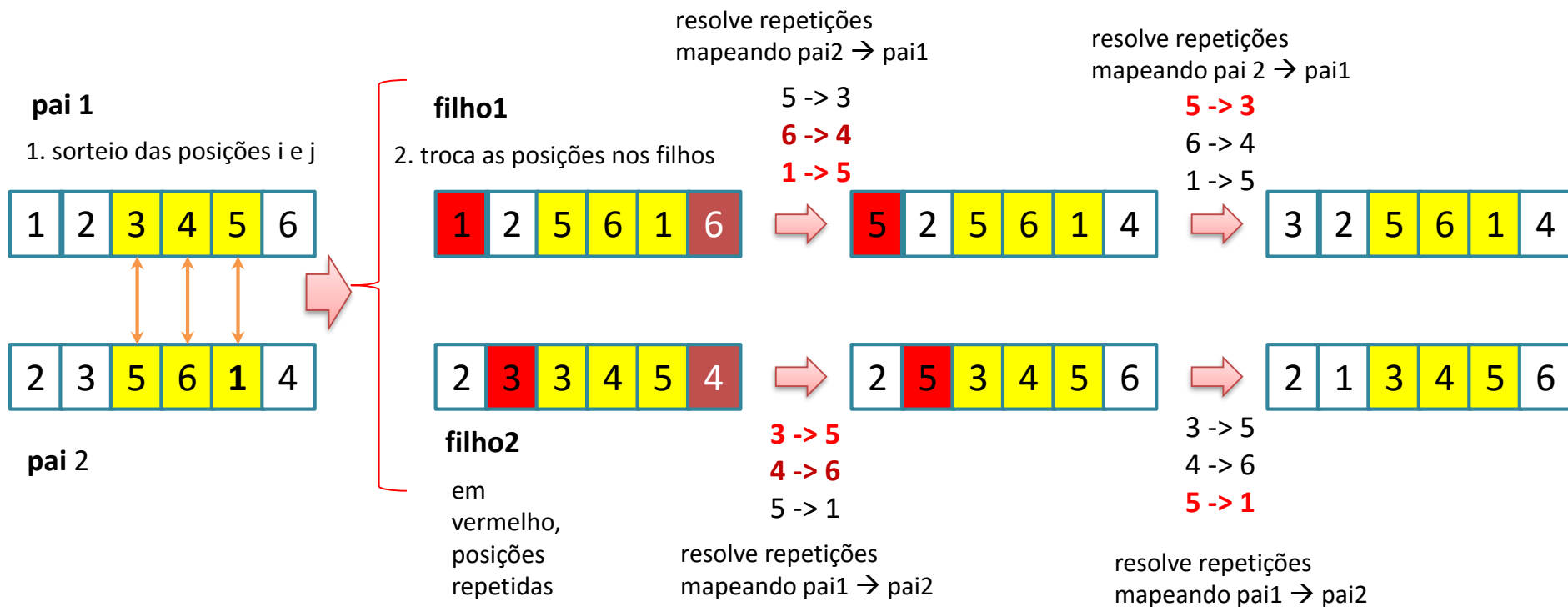
gene=id da cidade alelo={ids das cidades}

Caixeiro viajante

cidade	G	B	I	D	A	F	E	I	C	= cromossomo
ordem visita	1	2	3	4	5	6	7	8	9	

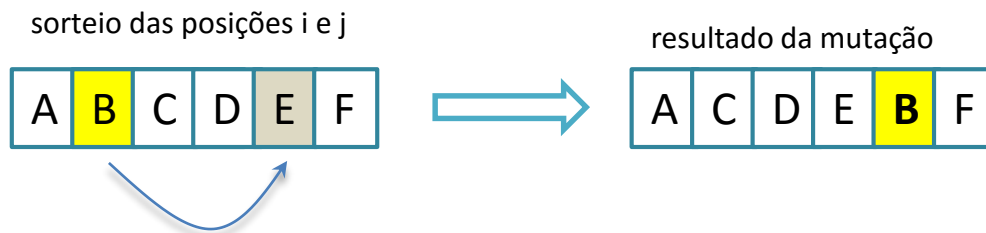
AG em Problemas de Permutação: tipos de crossover

Crossover de Mapeamento Parcial (PMX)

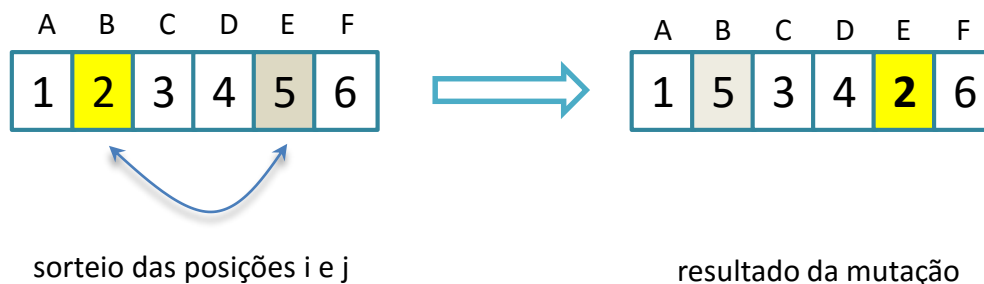


AG em Problemas de Permutação: tipos de mutação

Mutação baseada em posição:
retira locus da posição i e insere em j



Mutação baseada em ordem:
troca alelo (valor) da posição i com o da j



AG em Problemas de Permutação: tipos de mutação

Mutação por embaralhamento

Selecionar aleatoriamente uma sublista e embaralhá-la.

