Busca em Ambientes Complexos

Profa. Mercedes Gonzales Márquez

Tópicos

- Problemas de busca e otimização locais
  - Busca de subida de encosta (hill climbing)
  - Têmpera Simulada
  - Busca em feixe local
  - Algoritmos Genéticos

Busca de têmpera simulada (simulated annealing)

- Em metalurgia a têmpera é o processo usado para temperar ou endurecer metais e vidro aquecendo-os a alta temperatura e depois esfriando-os gradualmente, permitindo assim que o material alcance um estado de baixa energia. O processo começa em uma alta temperatura e prossegue com um resfriamento até atingir uma estabilidade.
- O algoritmo têmpera simulada permite algumas escolhas “ruins” para escapar de máximos locais. Mas vai decrescendo o número de escolhas ruins ao longo do tempo

Busca de têmpera simulada (simulated annealing)

- O algoritmo é similar a subida de encosta, porém, em vez de escolher o melhor movimento, ele escolhe um movimento aleatório. Se o movimento melhorar a situação, ele sempre será aceito. Caso contrário, o algoritmo aceitará o movimento com alguma probabilidade menor que 1. A probabilidade é reduzida exponencialmente com a “má qualidade” do movimento ou à medida que a “temperatura” T é reduzida.
- Movimentos “ruins” tem a probabilidade de serem permitidos no início, quando T estiver alto, e se tornam mais improváveis conforme T diminui.

Busca de têmpera simulada (simulated annealing)

Busca de têmpera simulada : Estado, Energia e Vizinhança

s

Estado candidato

Representa uma configuração possível do problema. Na busca, o algoritmo sempre mantém um estado atual.

E(s)

Energia

É a função de custo a minimizar. Estados com energia menor são considerados melhores.

N(s)

Vizinhança

É o conjunto de estados obtidos por uma alteração pequena, como mover uma única rainha.

- No exemplo das 8-rainhas

Estado	Energia	Vizinho
vetor $[r_1,\ldots,r_8]$, com uma rainha por coluna.	h é o número de pares de rainhas que se atacam.	mover uma rainha para outra linha na mesma coluna.

Busca de têmpera simulada : Regra de Aceitação

- A têmpera simulada sempre favorece soluções melhores, mas permite aceitar pioras com uma probabilidade que diminui quando a piora aumenta ou quando a temperatura cai.

MOVIMENTO PIOR

$$P = e^{-\Delta E/T}$$

$$\Delta E = E(\text{novo}) - E(\text{atual})$$

Em problemas de minimização, energia menor significa solução melhor. Nas 8 rainhas, usaremos **energia = h**.

↓	$\Delta E < 0$ o vizinho melhora	Aceitar o movimento, pois a energia diminui.
=	$\Delta E = 0$ o vizinho empata	Aceitar ou sortear, conforme a implementação adotada.
↑	$\Delta E > 0$ o vizinho piora	Aceitar com probabilidade P = $e^{-\Delta E/T}$.

Busca de têmpera simulada para problema 8-rainhas

A mesma função de avaliação usada no hill climbing aparece aqui como energia: menor h significa menos conflitos, mas o algoritmo pode aceitar pioras temporárias

Estado	Um vetor de oito números, sem precisar testar linhas vazias ou colunas repetidas.
Energia	$E(s)=h(s)$, o número de pares de rainhas que se atacam na mesma linha ou diagonal.
Vizinho	Escolhe-se uma coluna e uma nova linha. Existem $8\times7=56$ movimentos possíveis.
Amostra	Em vez de examinar os 56 a cada iteração, a têmpera simulada sorteia um candidato e decide pela regra probabilística.

$h(s)$	Número de pares de rainhas que se atacam. O objetivo é minimizar até $h=0$.
ΔE	Diferença entre a energia do candidato e a energia atual. Valor negativo é melhora.

$$P = e^{-\Delta E/T}$$

Se o movimento piora a solução, ele ainda pode ser aceito quando $u \leq P$.

Busca de têmpera simulada : Algoritmo para problema 8-rainhas

No problema das 8 rainhas, a têmpera simulada não avalia todos os sucessores a cada passo: ela sorteia um vizinho, calcula a nova energia e decide aceitar ou rejeitar probabilisticamente

```
s ← [4,5,4,3,4,5,5,4]
T ← 8,0 // temperatura inicial
enquanto h(s) > 0 e T > ε:
    sortear uma coluna c
    sortear uma nova linha r
    s' ← mover rainha de c para r
    ΔE ← h(s') − h(s)
    se ΔE ≤ 0: aceitar s'
    senão:
        P ← exp(−ΔE/T)
        u ← número aleatório em [0,1]
        se u ≤ P: aceitar s'
    T ← 0,92 · T
```

Busca de têmpera simulada : Regra de aceitação

Por que aceitar pioras?

- A têmpera simulada evita ficar presa em **mínimos locais.**

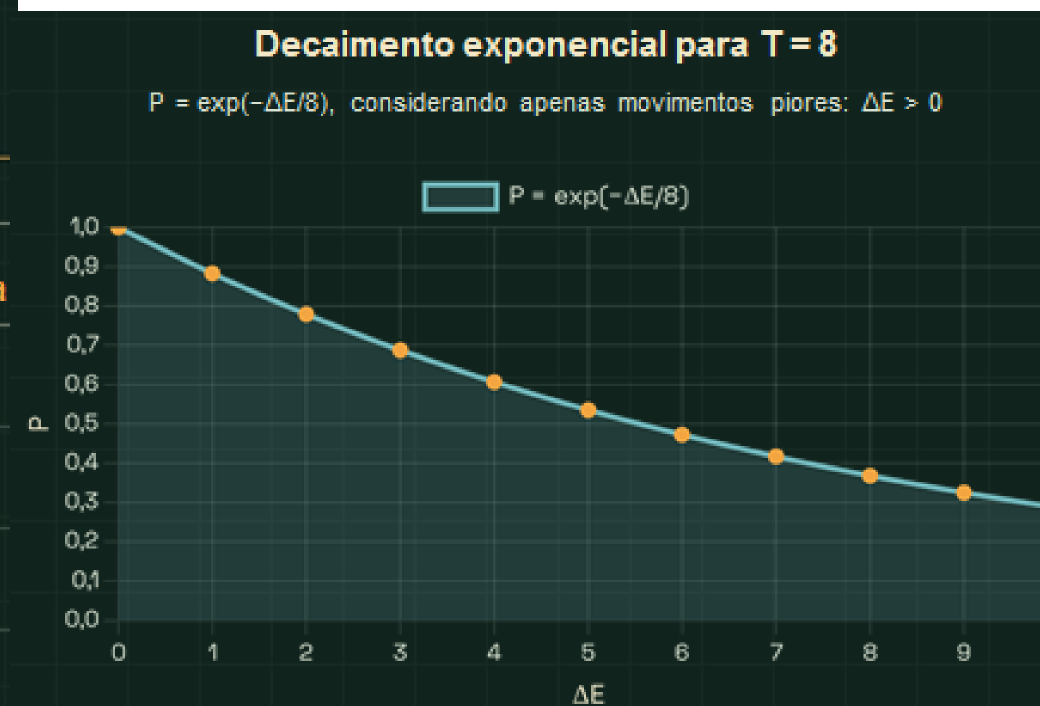
- Uma busca puramente gulosa rejeita qualquer movimento que aumente o custo. Isso pode bloquear a busca em uma solução que é boa apenas localmente. A probabilidade de aceitação permite aceitar algumas pioras controladas, atravessando regiões temporariamente ruins para alcançar soluções melhores depois.

Busca de têmpera simulada : Regra de aceitação

- **Quando ΔE aumenta, P cai:** Com a temperatura fixa, uma piora maior torna o expoente **$-\Delta E/T$** mais negativo; por isso a aceitação cai de forma exponencial.

$$P = e^{-\Delta E/8}$$

ΔE	Interpretação	P aproximada
1	piora pequena	0,882
2	piora moderada	0,779
5	piora relevante	0,535
10	piora grande	0,287



Busca de têmpera simulada : Regra de aceitação

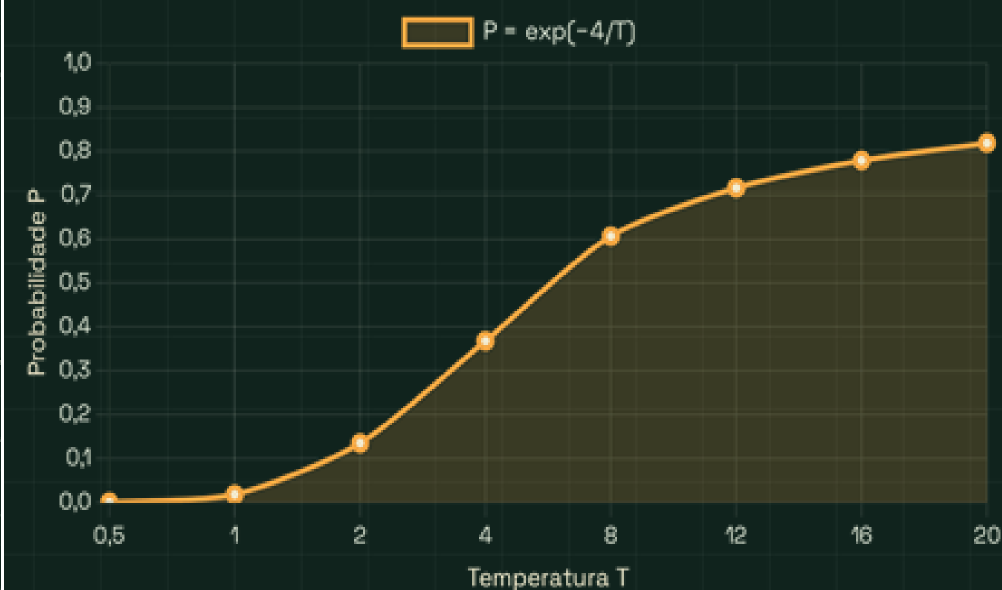
- **Quando T diminui, P cai:** Com a piora fixa, o resfriamento torna a aceitação cada vez menos provável

$$\Delta E = 4 \Rightarrow P = e^{-4/T}$$

Mantendo a mesma piora, a temperatura funciona como um controle de tolerância. Temperaturas altas suavizam a penalização; temperaturas baixas tornam o expoente muito negativo.

Temperatura T	Cálculo	P aproximada
20	$e^{-4/20}$	0,819
8	$e^{-4/8}$	0,607
2	$e^{-4/2}$	0,135
0,5	$e^{-4/0,5}$	0,0003

Curva de aceitação para uma piora de ΔE = 4

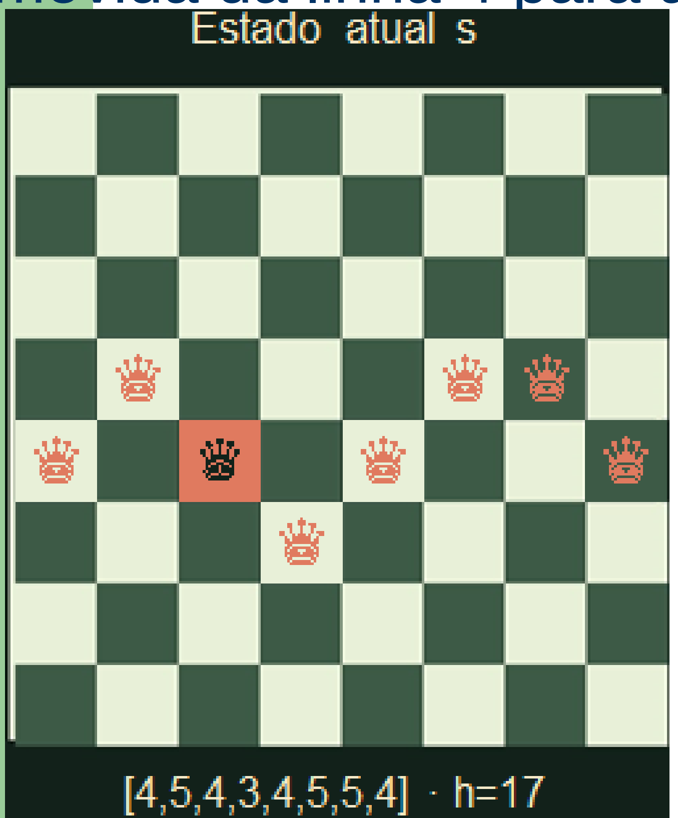


Leitura: no início quente, aceitar uma piora ainda é plausível; ao esfriar, a busca se aproxima de uma estratégia gulosa.

Problema 8-rainhas

Passo 1: sorteia um vizinho e calcula ΔE

Começo em h=17. No primeiro sorteio, a rainha da coluna 3 é movida da linha 4 para a linha 6, formando um candidato melhor.

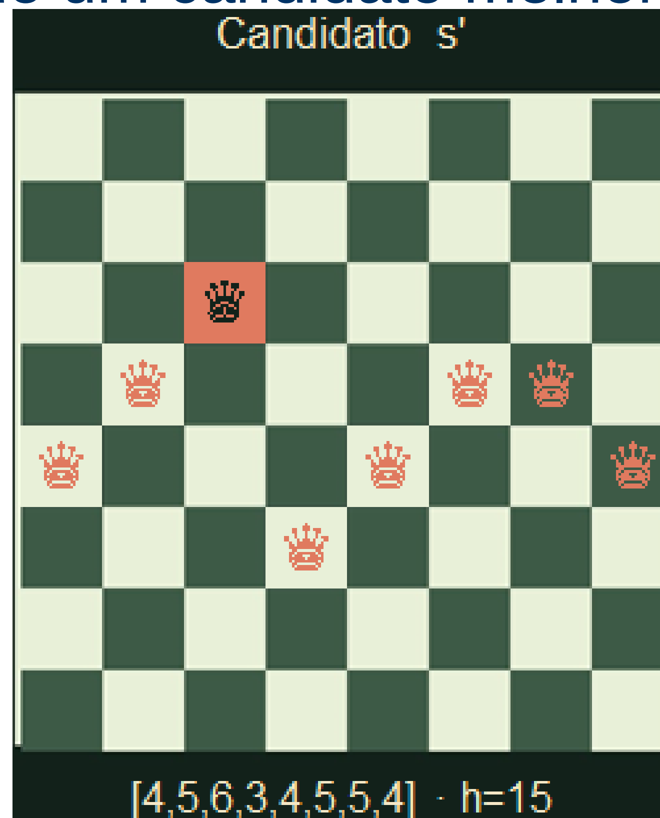


c3: 4 → 6

h(s') = 15

ΔE = 15 − 17 = −2

Como ΔE ≤ 0, o candidato reduz conflitos e é aceito sem depender de probabilidade



Problema 8-rainhas:
Primeiros passos: a energia cai de 17 para 3

Passo	T	Movimento	h→h'	ΔE	$P_{aceit.}$	u	Decisão
1	8,000	c3: 4→6	17→15	−2	1,000*	0,8474	aceita
2	7,360	c2: 5→3	15→12	−3	1,000*	0,1179	aceita
3	6,771	c8: 4→5	12→11	−1	1,000*	0,6516	aceita
4	6,229	c4: 3→1	11→11	0	1,000	0,4879	aceita
5	5,731	c7: 5→4	11→14	+3	0,5925	0,6074	rejeita
6	5,273	c1: 4→7	11→7	−4	1,000*	0,4454	aceita
7	4,851	c4: 1→6	7→6	−1	1,000*	0,9453	aceita
8	4,463	c6: 5→1	6→3	−3	1,000*	0,0223	aceita

Após o passo 8: Estado [7,3,6,6,4,1,5,5] e energia h=3

Problema 8-rainhas
O ponto-chave: aceitar algumas pioras

Passo	T	h→h'	ΔE	P	u	Decisão
5	5,731	11→14	+3	0,5925	0,6074	rejeita: u>P
9	4,106	3→6	+3	0,4816	0,6865	rejeita: u>P
12	3,197	2→4	+2	0,5350	0,2331	aceita: u≤P
13	2,941	4→6	+2	0,5066	0,2188	aceita: u≤P

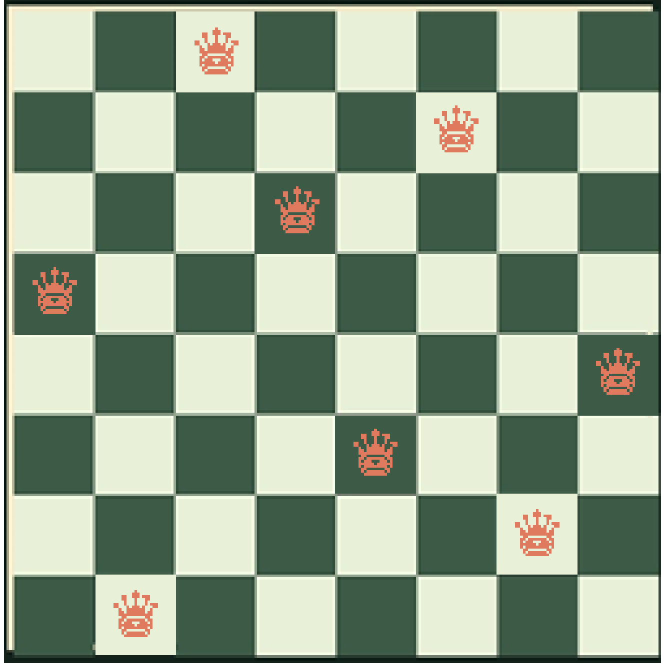
Interpretação: nos passos 12 e 13, a busca deliberadamente sai de uma região com h=2 e passa por h=4 e h=6. Essa liberdade temporária é o mecanismo que reduz o risco de ficar presa cedo demais.

Problema 8-rainhas
Resultado: solução encontrada com h=0

Marcos da execução

Momento	h	Estado
Inicial	17	[4,5,4,3,4,5,5,4]
Após passo 8	3	[7,3,6,6,4,1,5,5]
Após passo 11	2	[7,3,6,8,4,1,7,5]
Piora aceita no passo 12	4	[7,3,6,8,4,1,7,6]
Piora aceita no passo 13	6	[7,3,6,6,4,1,7,6]
Último passo aceito, 102	0	[5,1,8,6,3,7,2,4]

Tabuleiro final



- A execução não foi uma descida monotônica. A têmpera simulada combina descidas locais com aceitação controlada de pioras. Essa combinação amplia a exploração no início e concentra a busca quando a temperatura já está baixa.

Busca em feixe local

- Guarda na memória um número k de nós/estados gerados aleatoriamente.
- Em cada passo são gerados todos os sucessores de todos os k estados. Se qualquer um deles for o objetivo o algoritmo para. Caso contrário ele selecionará os k melhores sucessores a partir da lista completa e repetirá o procedimento.

Busca em feixe local

- **O parâmetro k controla exploração**
- Em feixe local, **k** define quantos estados sobrevivem a cada iteração: quanto maior o feixe, maior a diversidade explorada e maior o custo.

k = 1

Busca quase linear

O método se aproxima da subida de encosta, seguindo apenas o melhor caminho local disponível.

Risco: ficar preso rapidamente em um ótimo local.

k moderado

Equilíbrio prático

Mantém alternativas suficientes para escapar de escolhas ruins sem multiplicar demais as avaliações.

Risco: um k mal escolhido pode reduzir diversidade ou desperdiçar custo.

k alto

Exploração ampla

Investiga mais regiões do espaço de busca e reduz dependência de estados iniciais específicos.

Risco: custo computacional elevado por iteração.

Busca em feixe local: Exemplo simples

maximizar $f(x)=-(x-7)^2+49$

Considere valores inteiros de **x** entre 0 e 10. A função cresce quando o estado se aproxima de **x=7**, onde ocorre o máximo.

k	Manter apenas 2 estados por iteração.
N	Gerar vizinhos por x-1 e x+1.
f	Selecionar os sucessores com maior valor de aptidão.

Iteração	Estados			Sucessores avaliados			
0	x=1	x=9	→	0:0	2:24	8:48	10:40
1	x=8	x=10	→	7:49	9:45	9:45	
2	x=7	x=9	→	6:48	8:48	10:40	

melhor ponto: x=7, f(x)=49

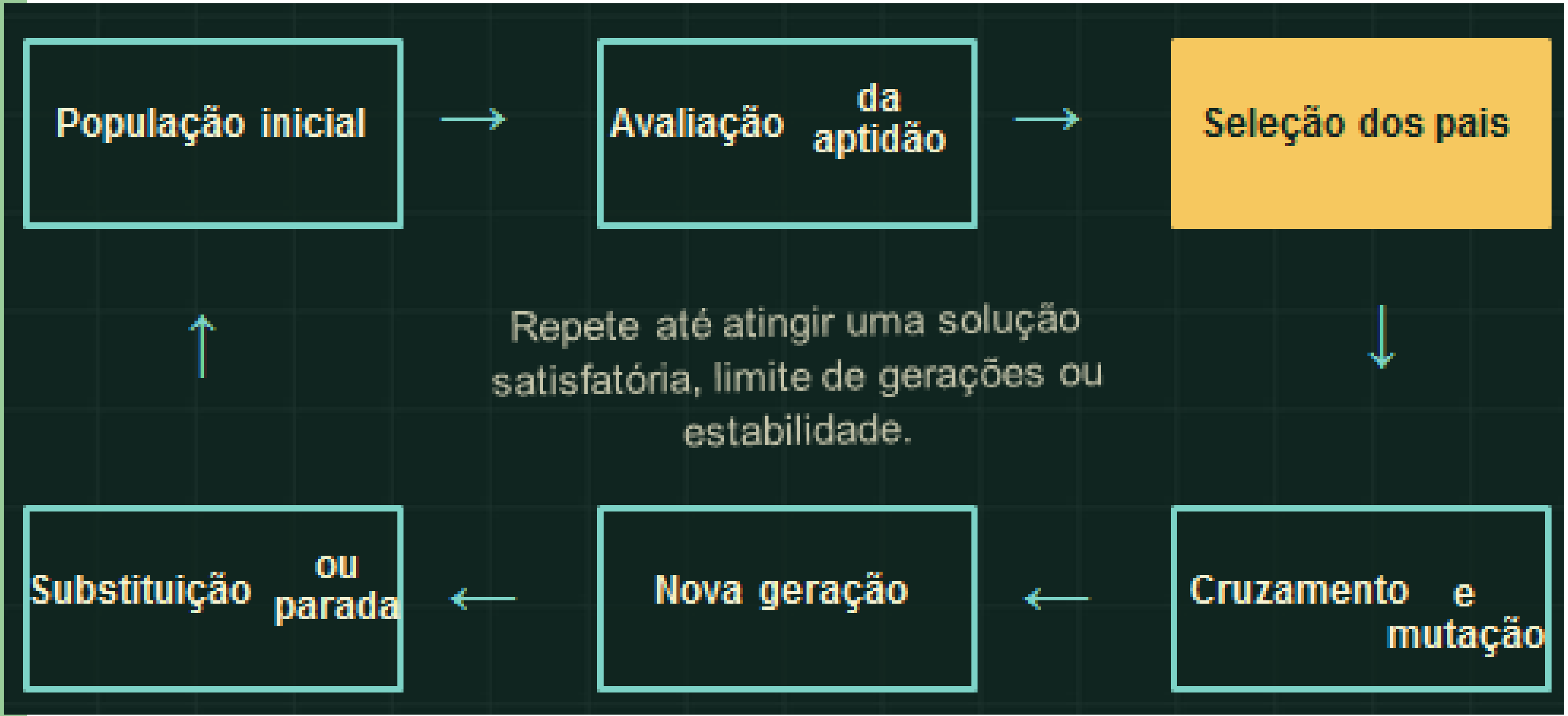
Algoritmos Genéticos

- Variante de Busca em Feixe Local estocástica.
- Algoritmos genéticos são métodos de busca inspirados na **seleção natural**: soluções melhores têm maior chance de gerar novas soluções.

Indivíduo	Uma solução candidata dentro da população.
Cromossomo	A representação codificada da solução, como bits, números ou permutações.
Aptidão	Medida que indica o quanto a solução atende ao objetivo do problema.

Ciclo de um algoritmo genético

- Uma população evolui por gerações: soluções melhores tendem a **sobreviver** e recombinar características.

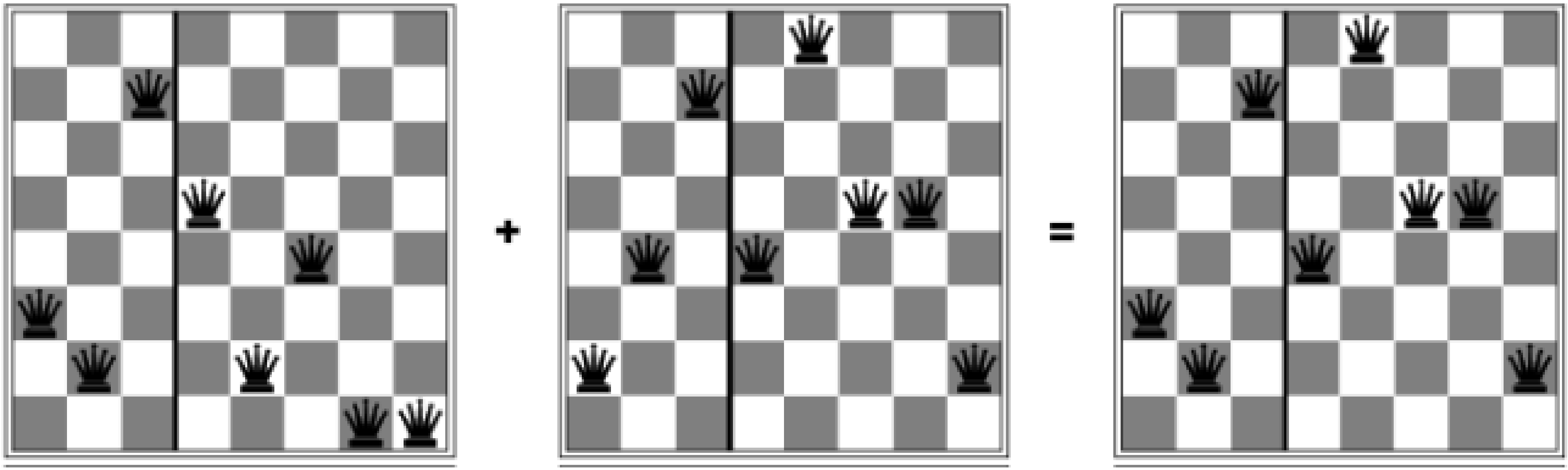


Algoritmos Genéticos – Problema das 8-rainhas

- Função de fitness: número de pares de rainhas que não estão se atacando (min=0,max=8×7/2=28). 24/(24+23+20+11)=31 23/(24+23+20+11)=29, etc
- Considerando seleção por roleta escolhe-se 4 pares que seria (2,1) e (2,3)

Algoritmos Genéticos

- Os crossover pode ser significativo se as strings forem significativas.



Exercício

Considere a função fitness=sin(πx/256), 0<=x<=255. Use Algoritmos Genéticos para encontrar o ponto máximo da função. Use cromossomos binários de 8 digitos. Crie uma população inicial com 8 indivíduos. Use 7 gerações.

Exemplo de população inicial

Primeira Geração

Cromossomo do Filho	x	Fitness
01011001	89	0.887640
11000110	198	0.653173
00101001	41	0.482184
10000001	129	0.999925
11110111	247	0.110222
01011111	95	0.919114
10000001	129	0.999925
10000010	130	0.999699

Seleção (roleta) de 4 pares

Seleção e Cruzamento:
5×8 (Corte 6),
6×4 (Sem Cruz),
4×4 (Corte 1),
2×8 (Corte 7)

Exercício – Segunda Geração

ID	Origem	Cromossomos dos Pais	Cromossomo do Filho	x	Fitness
1	Elite (da Gen 1)	-- N/A --	10000001	129	0.999925
2	Pares 5×8 F1 (Corte 6)	P1 111101|11 P2 100000|10	111101 | 10	246	0.122411
3	Pares 5×8 F2 (Corte 6) + MUT(5)	P1 100000|10 P2 111101|11	10000 1 11	135	0.996313
4	Pares 6×4 F1 (Sem Cruz)	P1 01011111 P2 10000001	01011111	95	0.919114
5	Pares 6×4 F2 (Sem Cruz) + MUT(2	P1 10000001 P2 01011111	10 1 00001	161	0.919114
6	Pares 4×4 F1 (Corte 1)	P1 1|0000001 P2 1|0000001	1 | 0000001	129	0.999925
7	Pares 4×4 F2 (Corte 1) + MUT(0,3,	P1 1|0000001 P2 1|0000001	0 00 1 00 1 1	19	0.231058
8	Pares 2×8 F1 (Corte 7)	P1 1100011|0 P2 1000001|0	1100011 | 0	198	0.653173

Seleção (roleta) de 4 pares

5×1 (Corte 1),
1×3 (Corte 1),
5×8 (Sem Cruz),
5×4 (Corte 1)

Exercício – Terceira Geração

ID	Origem	Cromossomos dos Pais	Cromossomo do Filho	x	Fitness
1	Elite (da Gen 2)	-- N/A --	10000001	129	0.999925
2	Pares 5×1 F1 (Corte 1) + MUT(2)	P1 1|0100001 P2 1|0000001	10 1 00001	161	0.919114
3	Pares 5×1 F2 (Corte 1)	P1 1|0000001 P2 1|0100001	1 | 0100001	161	0.919114
4	Pares 1×3 F1 (Corte 1)	P1 1|0000001 P2 1|0000111	1 | 0000111	135	0.996313
5	Pares 1×3 F2 (Corte 1)	P1 1|0000111 P2 1|0000001	1 | 0000001	129	0.999925
6	Pares 5×8 F1 (Sem Cruz)	P1 10100001 P2 11000110	10100001	161	0.919114
7	Pares 5×8 F2 (Sem Cruz)	P1 11000110 P2 10100001	11000110	198	0.653173
8	Pares 5×4 F1 (Corte 1)	P1 1|0100001 P2 0|1011111	1 | 1011111	223	0.393992

Seleção (roleta) de 4 pares

7×7 (Corte 1),
2×1 (Corte 3),
1×8 (Corte 1),
2×7 (Corte 3)

Exercício – Quarta Geração

ID	Origem		Cromossomos dos Pais	Cromossomo do Filho	x	Fitness
1	Elite (da Gen 3)		-- N/A --	10000001	129	0.999925
2	Pares 7×7 F1 (Corte 1)	P1 P2	1|1000110 1|1000110	1 | 1000110	198	0.653173
3	Pares 7×7 F2 (Corte 1)	P1 P2	1|1000110 1|1000110	1 | 1000110	198	0.653173
4	Pares 2×1 F1 (Corte 3) + MUT(1)	P1 P2	101|00001 100|00001	1 1 100001	225	0.371317
5	Pares 2×1 F2 (Corte 3) + MUT(5)	P1 P2	100|00001 101|00001	10000 1 01	133	0.998118
6	Pares 1×8 F1 (Corte 1)	P1 P2	1|0000001 1|1011111	1 | 1011111	223	0.393992
7	Pares 1×8 F2 (Corte 1) + MUT(3)	P1 P2	1|1011111 1|0000001	100 1 0001	145	0.978317
8	Pares 2×7 F1 (Corte 3)	P1 P2	101|00001 110|00110	101 | 00110	166	0.893224

Seleção (roleta) de 4 pares

1×8 (Sem Cruz),
6×2 (Corte 6),
8×5 (Corte 5),
1×7 (Corte 1)

Exercício – Sétima Geração

ID	Origem	Cromossomos dos Pai:		Cromossomo do Filho	x	Fitness
1	Elite (da Gen 6)	-- N/A --		10000001	129	0.999925
2	Pares 8×6 F1 (Sem Cruz)	P1	10010001	10010001	145	0.978317
		P2	10100000			
3	Pares 8×6 F2 (Sem Cruz) + MUT(:	P1	10100000	10 0 00000	128	1.000000
		P2	10010001			
4	Pares 7×5 F1 (Corte 7) + MUT(0)	P1	1001010|1	0 0010101	21	0.254866
		P2	1010011|1			
5	Pares 7×5 F2 (Corte 7) + MUT(3)	P1	1010011|1	101 1 0111	183	0.780737
		P2	1001010|1			
6	Pares 5×5 F1 (Sem Cruz)	P1	10100111	10100111	167	0.887640
		P2	10100111			
7	Pares 5×5 F2 (Sem Cruz)	P1	10100111	10100111	167	0.887640
		P2	10100111			
8	Pares 1×6 F1 (Corte 1) + MUT(1)	P1	1|0000001	1 1 100000	224	0.382683
		P2	1|0100000			

O algoritmo atingiu o **ótimo global perfeito (x=128)** exatamente na Geração 7. O fitness alcançou o valor máximo teórico de 1.000000.

Exemplo completo das 8-rainhas

População de 4 indivíduos que geram 4 indivíduos em cada geração.

População Inicial e Avaliação

Ind.	Cromossomo	Conflitos	Aptidão
P1	[1,5,8,6,3,7,2,2]	2	26
P2	[4,2,7,3,6,8,5,2]	2	26
P3	[2,7,4,1,8,3,6,5]	8	20
P4	[1,2,3,4,5,6,7,8]	28	0

Seleção e Cruzamento geram 4 filhos

Selecionamos os dois pais de maior aptidão e usamos **2 pontos de corte** para produzir uma população candidata com quatro descendentes.

Filho	Corte	Composição	Cromossomo resultante	Aptidão
F1	após col. 4	P1 nas col. 1–4 + P2 nas col. 5–8	[1,5,8,6,6,8,5,2]	22
F2	após col. 4	P2 nas col. 1–4 + P1 nas col. 5–8	[4,2,7,3,3,7,2,2]	22
F3	após col. 6	P1 nas col. 1–6 + P2 nas col. 7–8	[1,5,8,6,3,7,5,2]	25
F4	após col. 6	P2 nas col. 1–6 + P1 nas col. 7–8	[4,2,7,3,6,8,2,2]	24

Mutação pontual nos quatro descendentes

Depois do cruzamento, **cada filho** pode sofrer uma mutação pequena. Aqui, cada linha altera apenas **um gene** do próprio descendente.

Filho	Antes da mutação	Gene alterado	Depois da mutação	Aptidão
F1	[1,5,8,6,6,8,5,2]	col. 5: 6→3	M1=[1,5,8,6,3,8,5,2]	23
F2	[4,2,7,3,3,7,2,2]	col. 8: 2→4	M2=[4,2,7,3,3,7,2,4]	24
F3	[1,5,8,6,3,7,5,2]	col. 8: 2→4	M3=[1,5,8,6,3,7,5,4]	25
F4	[4,2,7,3,6,8,2,2]	col. 8: 2→4	M4=[4,2,7,3,6,8,2,4]	26

A população inteira evolui com ciclo completo

Ger.	Indivíduo 1	Indivíduo 2	Indivíduo 3	Indivíduo 4
G0	P1 [1,5,8,6,3,7,2,2] fit 26	P2 [4,2,7,3,6,8,5,2] fit 26	P3 [2,7,4,1,8,3,6,5] fit 20	P4 [1,2,3,4,5,6,7,8] fit 0
G1	M1 [1,5,8,6,3,8,5,2] P1/P2: cruz. + mut. fit 23	M2 [4,2,7,3,3,7,2,4] P2/P1: cruz. + mut. fit 24	M3 [1,5,8,6,3,7,5,4] P1/P2: cruz. + mut. fit 25	M4 [4,2,7,3,6,8,2,4] P2/P1: cruz. + mut. fit 26
G2	Q1 [1,5,8,6,3,8,2,5] M3→M4 c5; col8 4→5 fit 25	Q2 [3,2,7,3,6,7,5,4] M4→M3 c5; col1 4→3 fit 23	Q3 [1,5,8,6,3,8,2,4] M3→M4 c4; col5 6→3 fit 26	Q4 [4,2,7,3,3,7,5,1] M4→M3 c4; col8 4→1 fit 25
G3	R1 [1,5,8,6,3,7,2,4] Q1→Q3 c5; col6 8→7 fit 28	R2 [1,5,8,6,3,7,2,5] Q3→Q1 c5; col6 8→7 fit 26	R3 [1,5,8,6,3,8,2,5] Q1→Q3 c4; col8 4→5 fit 25	R4 [1,5,7,6,3,8,2,5] Q3→Q1 c4; col3 8→7 fit 25

Solução válida das 8 rainhas