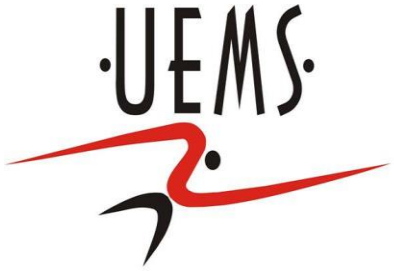


Busca em Ambientes Complexos

Profa. Mercedes Gonzales Márquez

Tópicos

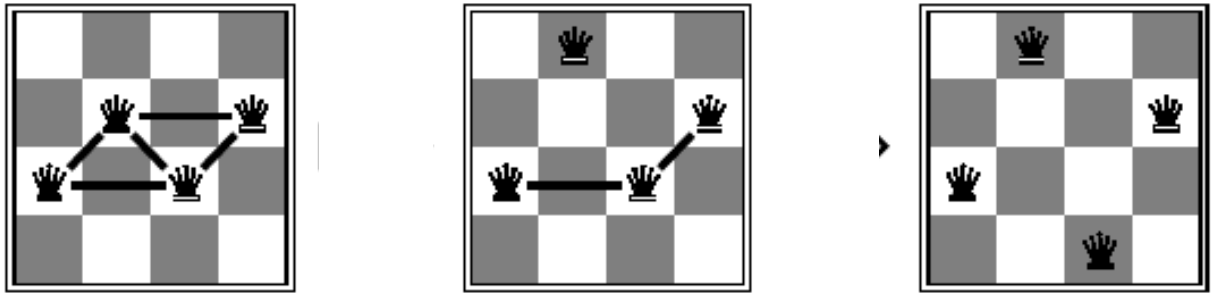
- Problemas de busca e otimização locais
  - Busca de subida de encosta (hill climbing)
  - Têmpera Simulada
  - Busca em feixe local
  - Algoritmos Genéticos

Problemas de busca e otimização locais

- Nos problemas de busca das aulas anteriores queríamos explorar caminhos pelo espaço de busca como, por exemplo, o caminho de Curitiba a Fortaleza.
- Às vezes só nos importamos com o estado final, o caminho até o objetivo é irrelevante.
- Em vez de armazenar todo o trajeto, um algoritmo de busca local mantém apenas o estado atual e realiza “movimentos” iterativos para estados vizinhos na tentativa de maximizar ou minimizar uma função de avaliação.
- Vantagens: Usam pouca memória e podem encontrar soluções razoáveis em grandes espaço de busca para os quais os algoritmos anteriormente vistos são inadequados.

Problemas as n rainhas

- Colocar n rainhas em um tabuleiro n×n, sendo que cada linha, coluna ou diagonal pode ter apenas uma rainha, ou seja não deverá haver um par de rainhas se atacando uma a outra.

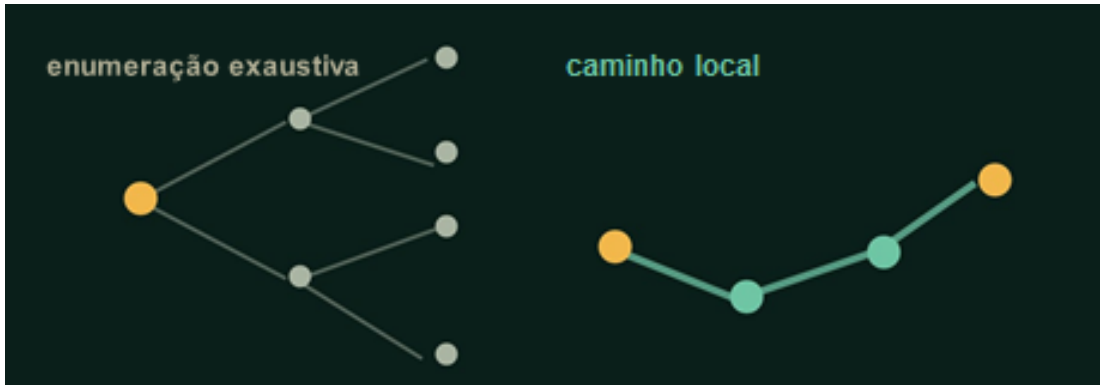


h = 5 h = 2 h = 0

- Veja que no primeiro exemplo temos 5 pares de rainhas se atacando uma a outra. No segundo, temos 2 pares de rainhas se atacando uma outra e só no último tabuleiro não temos nenhum par de rainhas atacando uma a outra.

Visão geral da busca local

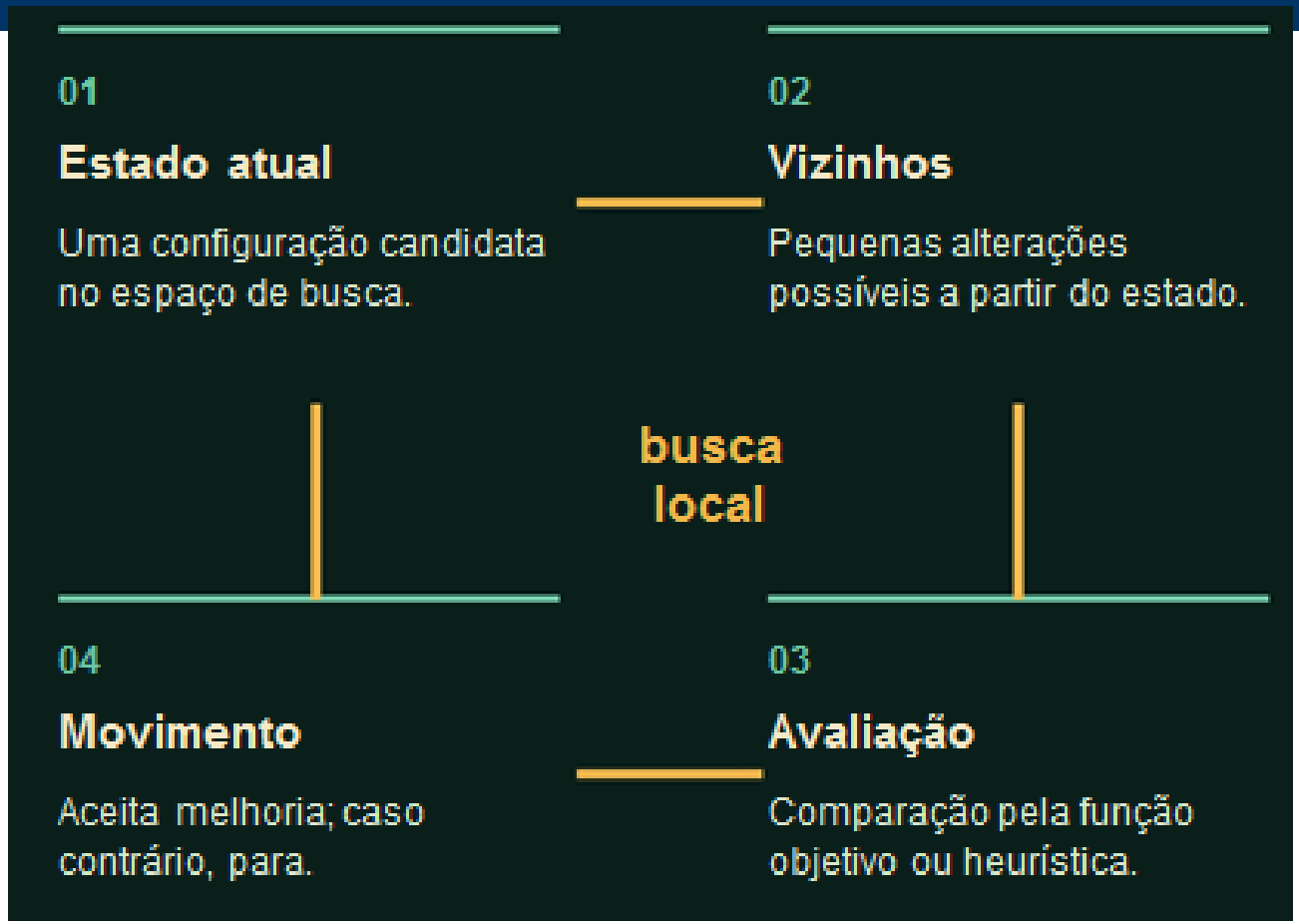
- Boas soluções aparecem quando exploramos vizinhos promissores.
- Inicia com uma **solução candidata** e aplica pequenas mudanças sucessivas.
- Percorre um **caminho único**, não uma árvore completa de possibilidades.



- É útil para grandes espaços de busca

Ciclo operacional da busca local

enumeração exaustiva caminho local



01

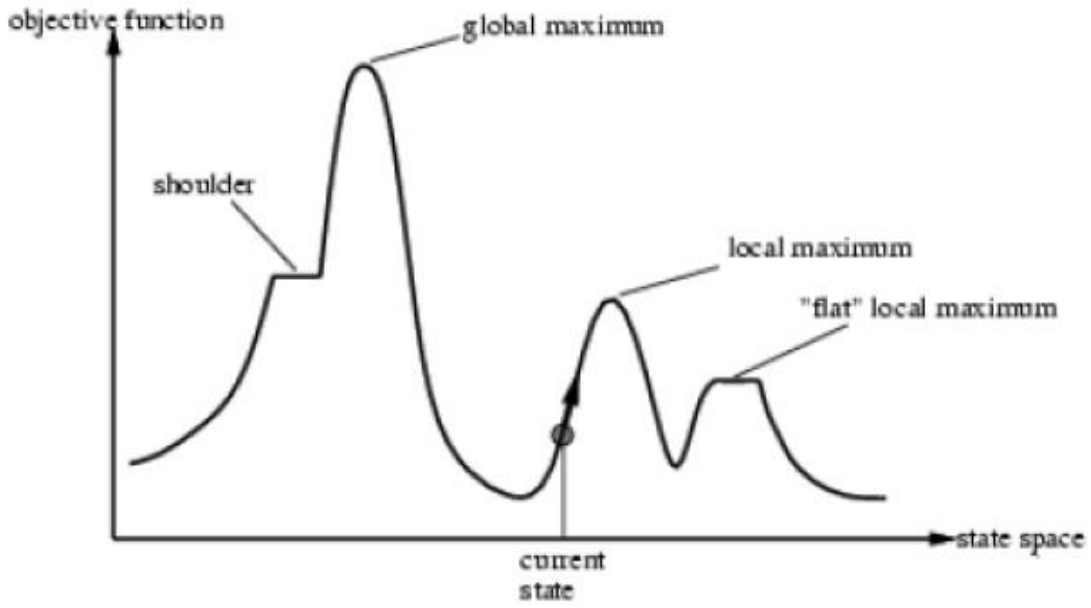
02

03

Busca de subida de encosta

- O algoritmo consiste em um laço repetitivo que percorre o espaço de estados no sentido do valor crescente, ou seja, ele segue na direção que oferece **a encosta mais íng**reme.
- Termina quando encontra um pico em que nenhum vizinho tem valor mais alto.
- A subida de encosta não examina antecipadamente valores de estados além dos vizinhos imediatos do estado atual.

Busca de subida de encosta

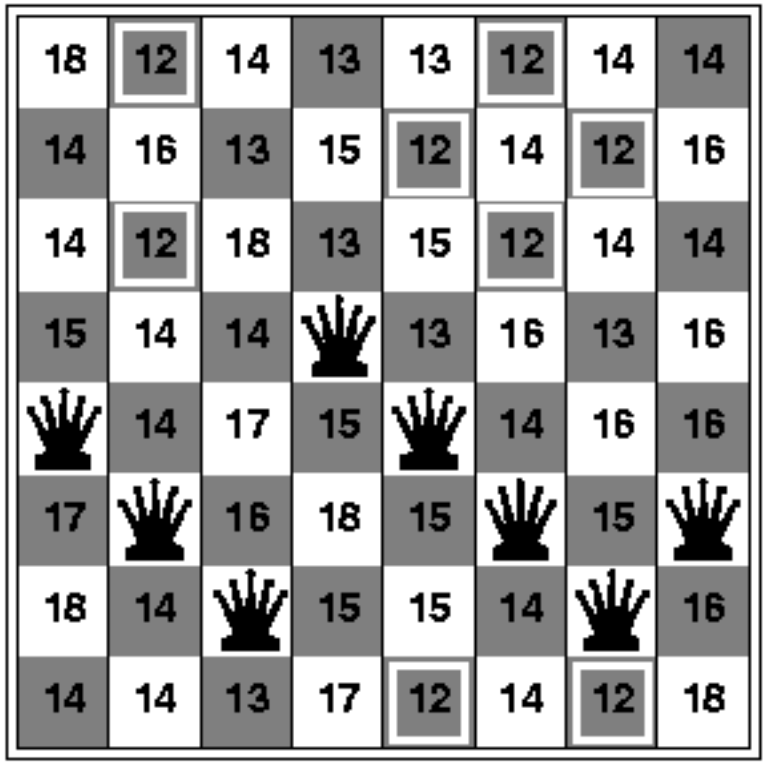


- Uma forma de usar a busca de subida de encosta é utilizar o negativo de uma função de custo heurística como função objetivo.

Busca de subida de encosta

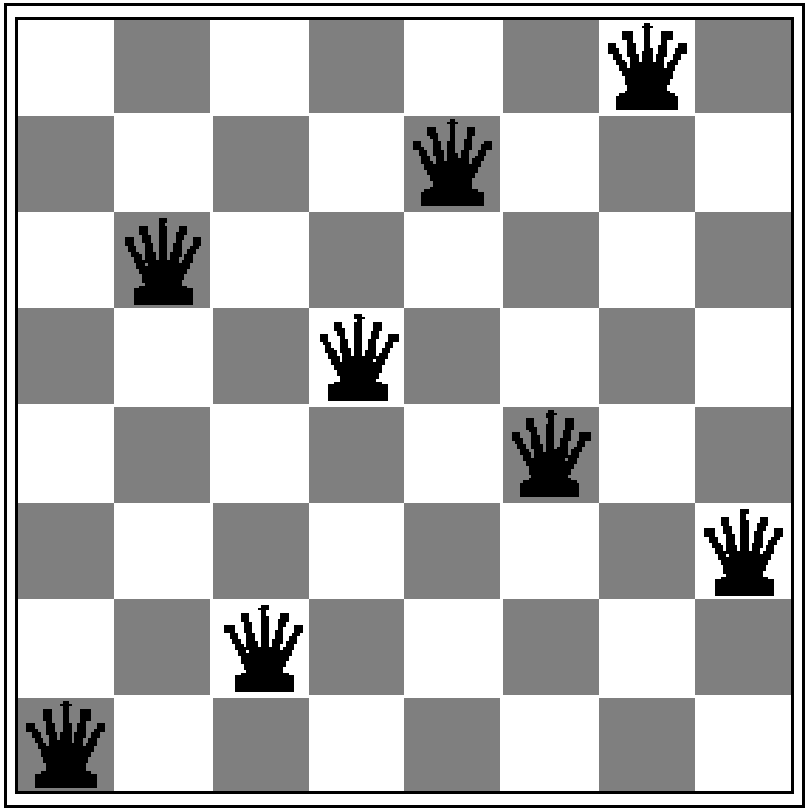
- Exemplo das 8 rainhas.
- Cada estado tem oito rainhas no tabuleiro, uma por coluna.
- Estado inicial escolhido aleatoriamente e
- Todos os sucessores de um estado são estados possíveis gerados pela movimentação de uma única rainha para outro quadrado na mesma coluna (de forma que cada estado tenha 8 x 7 = 56 sucessores).
- A função de custo heurística h é o número de pares de rainhas que estão se atacando umas às outras.

Busca de subida de encosta



- h=17 para o estado ao lado.
- O tabuleiro mostra o valor de h para cada sucessor possível obtido pela movimentação de uma rainha dentro da sua coluna.
- Existem 8 movimentos que estão empatados como melhores, com h=12. O algoritmo de subida de encosta escolherá um deles.

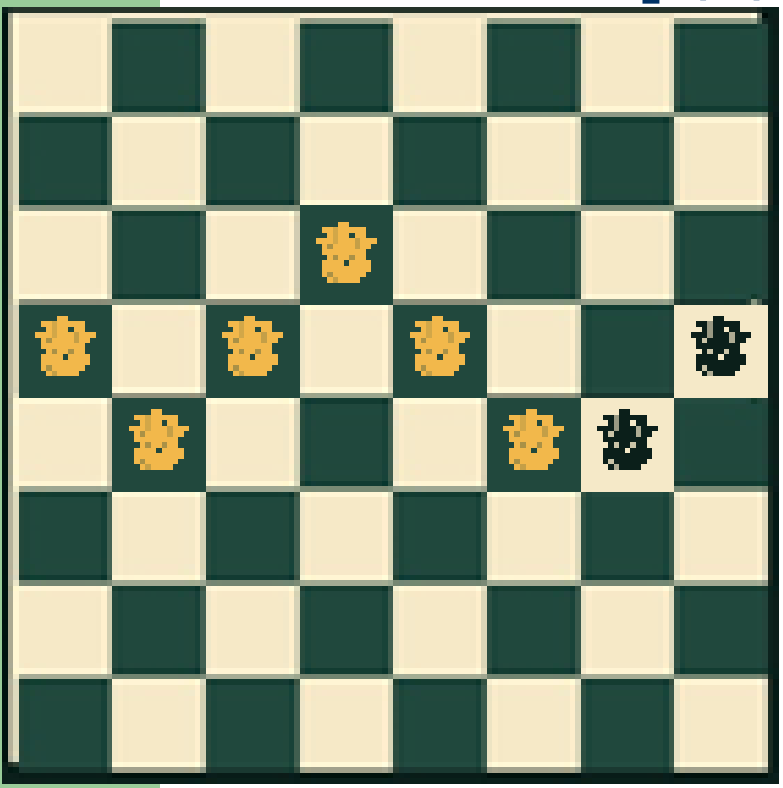
Busca de subida de encosta



- O estado tem h=1, mas qualquer movimento de qualquer rainha só piora a situação...
- Logo é um **máximo local**! (mínimo local para h)

Exemplo completo

- Estado inicial: h=17
- **Estado atual: [4,5,4,3,4,5,5,4]**



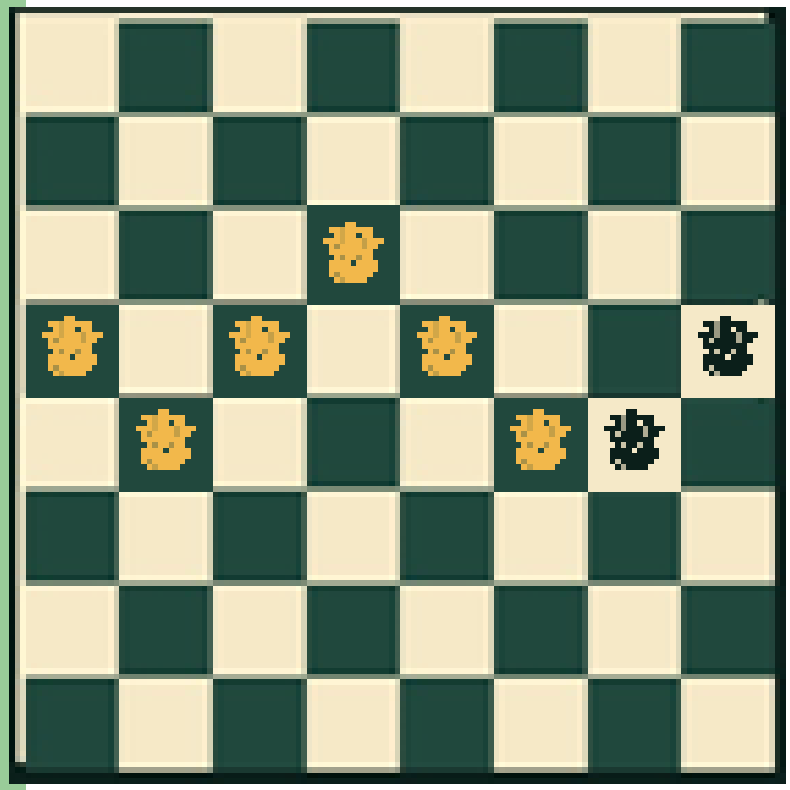
h(s)=17 Próximo estado: [4,2,4,3,4,5,5,4], h=12

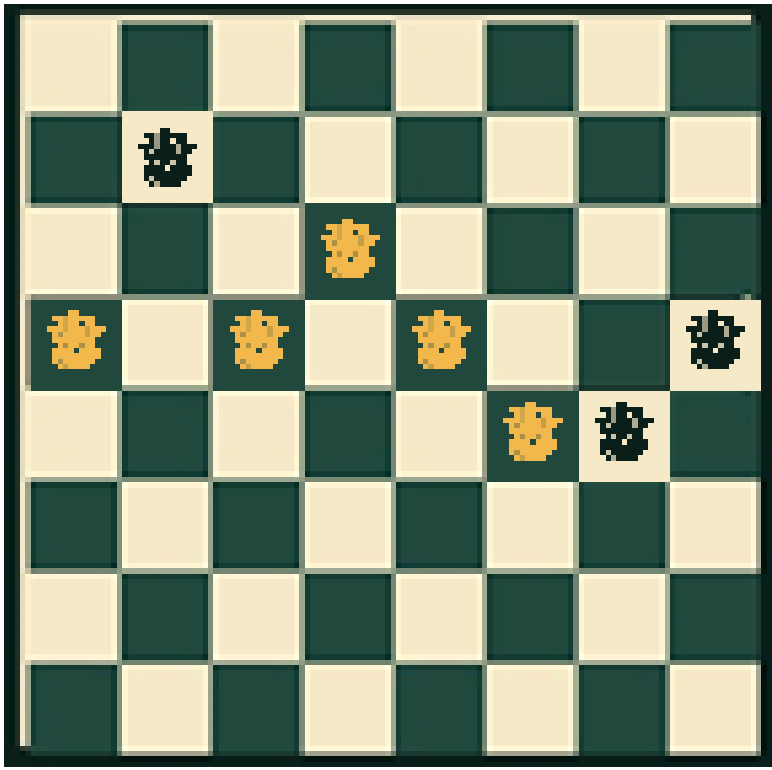
Todos os 56 sucessores: linha de destino › h

opção	c1	c2	c3	c4	c5	c6	c7	c8
1	1→13	1→15	1→13	1→14	1→13	1→16	1→14	1→14
2	2→14	2→12	2→16	2→14	2→15	2→14	2→15	2→13
3	3→14	3→15	3→13	4→17	3→14	3→15	3→16	3→15
4	5→16	4→16	5→15	5→18	5→15	4→18	4→19	5→16
5	6→16	6→12	6→15	6→13	6→14	6→15	6→17	6→14
6	7→13	7→13	7→12	7→16	7→14	7→14	7→14	7→16
7	8→14	8→12	8→13	8→15	8→14	8→13	8→15	8→13

Exemplo completo

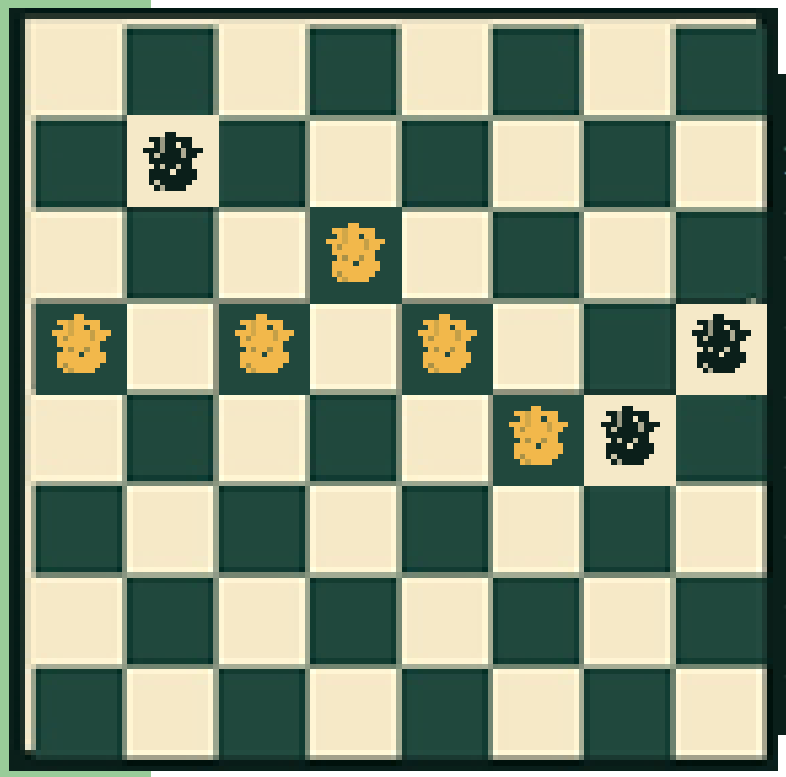
- melhor movimento c2: 5→2, h=12





Exemplo completo

- **Estado atual: [4,2,4,3,4,5,5,4], h=12**



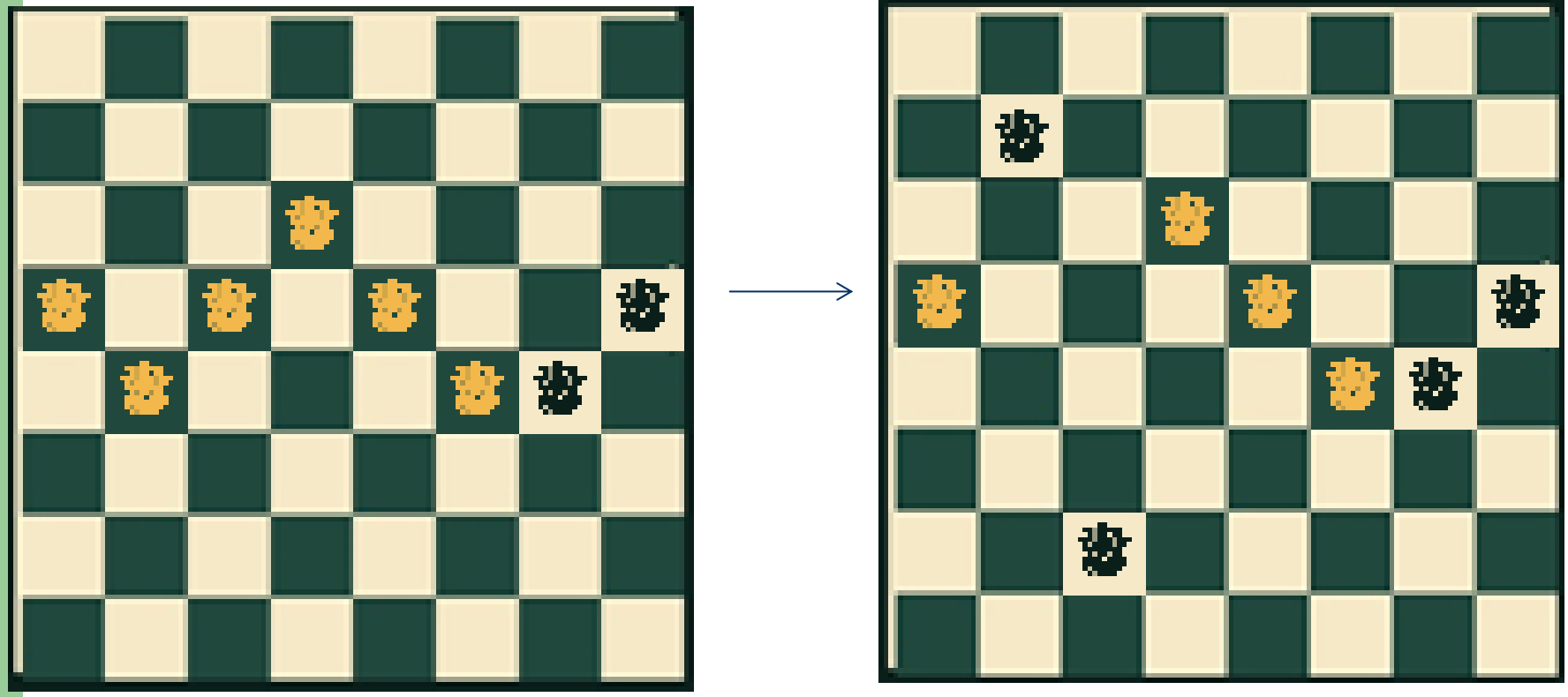
h(s)=12 Próximo estado: [4,2,7,3,4,5,5,4], h=8

Todos os 56 sucessores: linha de destino › h

opção	c1	c2	c3	c4	c5	c6	c7	c8
1	1→10	1→15	1→10	1→10	1→8	1→11	1→10	1→9
2	2→11	3→15	2→13	2→11	2→10	2→11	2→12	2→9
3	3→11	4→16	3→10	4→14	3→9	3→11	3→12	3→10
4	5→11	5→17	5→10	5→13	5→10	4→14	4→15	5→10
5	6→11	6→12	6→10	6→9	6→9	6→12	6→13	6→9
6	7→9	7→13	7→8	7→11	7→9	7→10	7→11	7→11
7	8→10	8→12	8→9	8→11	8→8	8→9	8→11	8→9

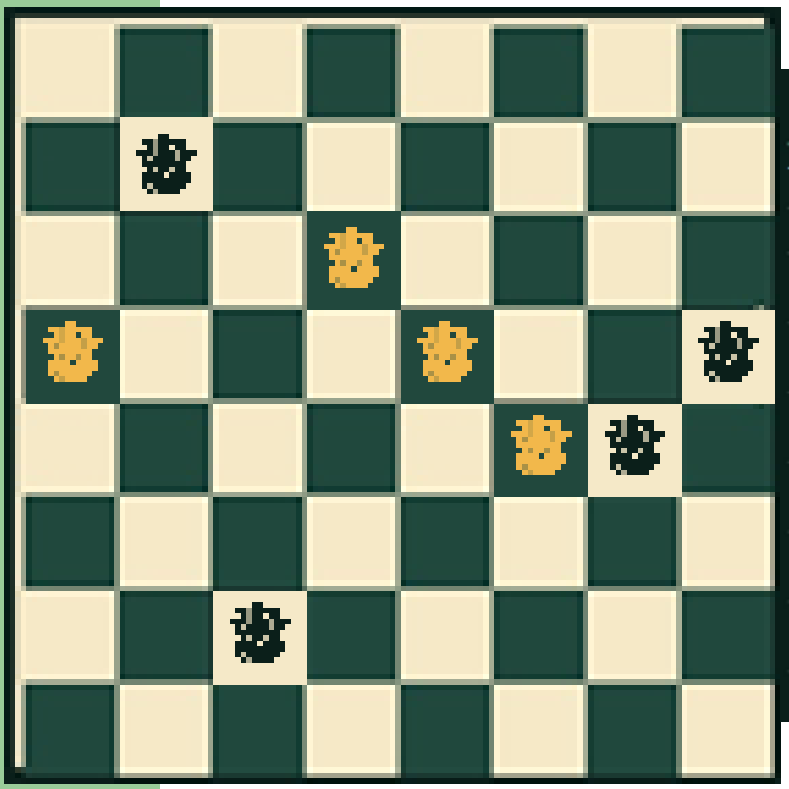
Exemplo completo

- melhor movimento c3: 4→7, h=8



Exemplo completo

- **Estado atual: [4,2,7,3,4,5,5,4], h=8**



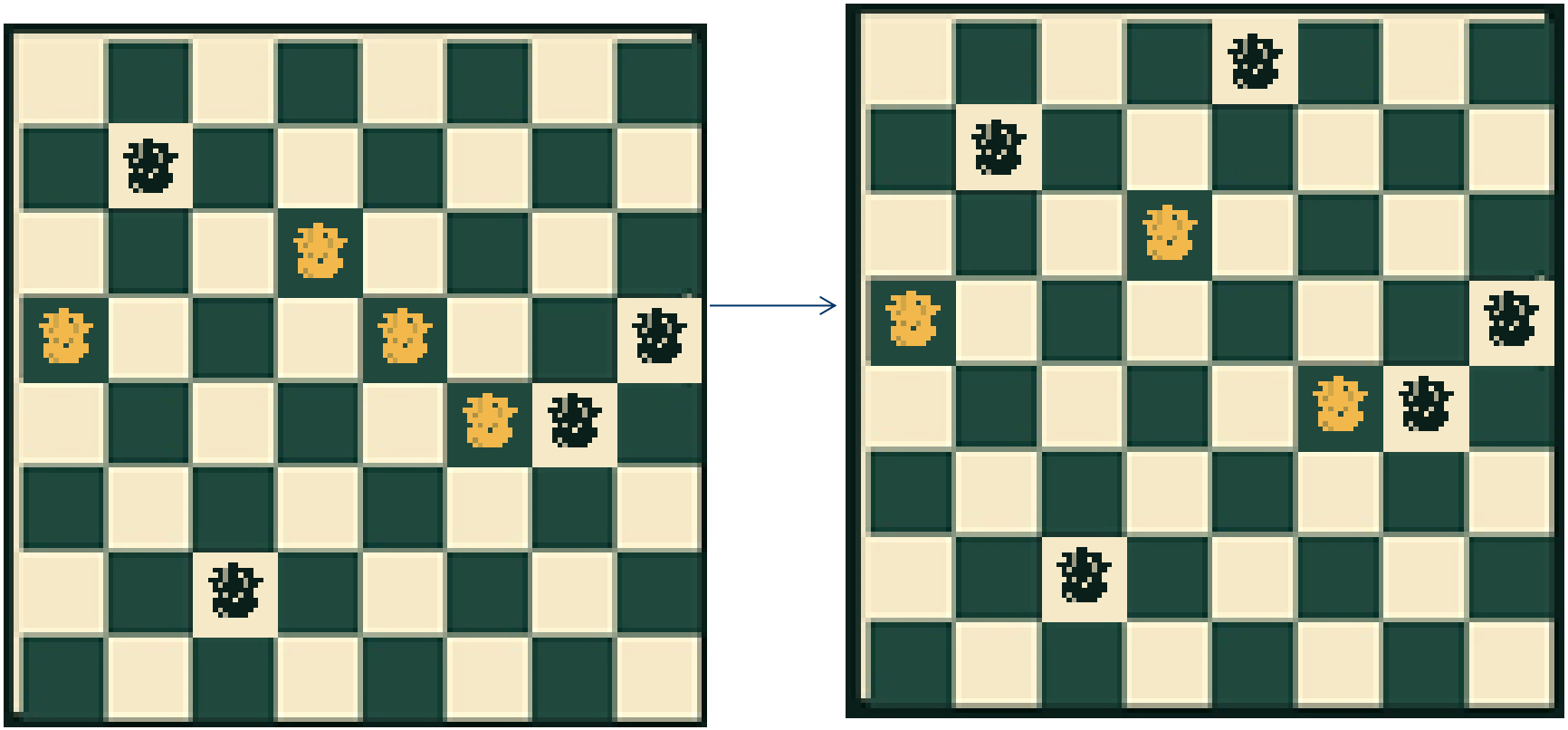
h(s)=8 Próximo estado: [4,2,7,3,1,5,5,4], h=5

Todos os 56 sucessores: linha de destino › h

opção	c1	c2	c3	c4	c5	c6	c7	c8
1	1→7	1→11	1→10	1→7	1→5	1→6	1→6	1→6
2	2→7	3→10	2→13	2→8	2→6	2→7	2→8	2→7
3	3→8	4→11	3→10	4→10	3→6	3→7	3→9	3→7
4	5→9	5→12	4→12	5→9	5→8	4→10	4→10	5→7
5	6→7	6→9	5→10	6→7	6→5	6→8	6→9	6→6
6	7→7	7→10	6→10	7→9	7→7	7→6	7→8	7→9
7	8→7	8→9	8→9	8→9	8→5	8→5	8→6	8→6

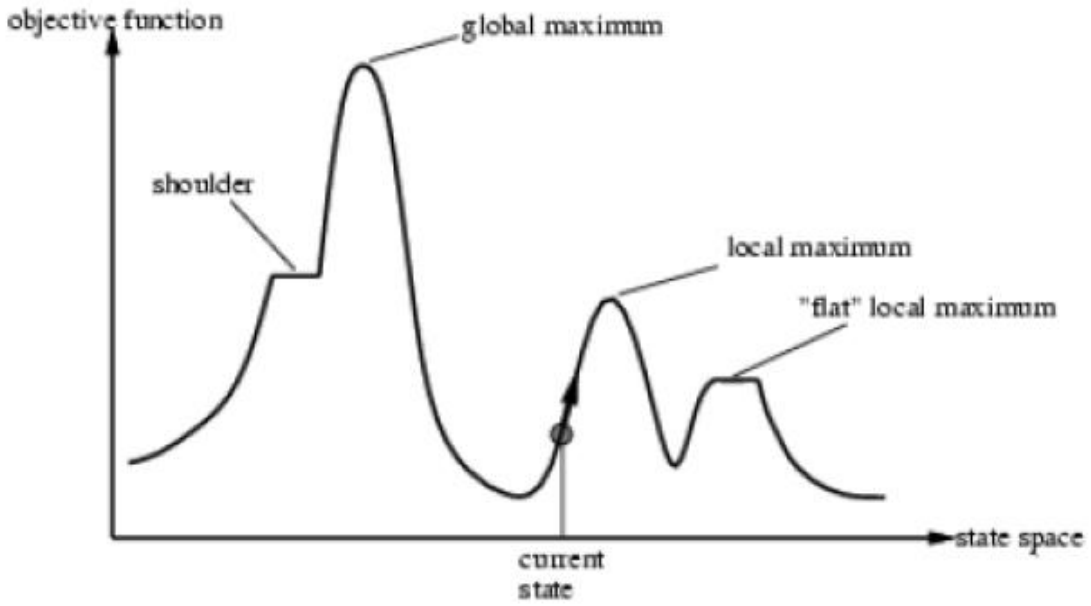
Exemplo completo

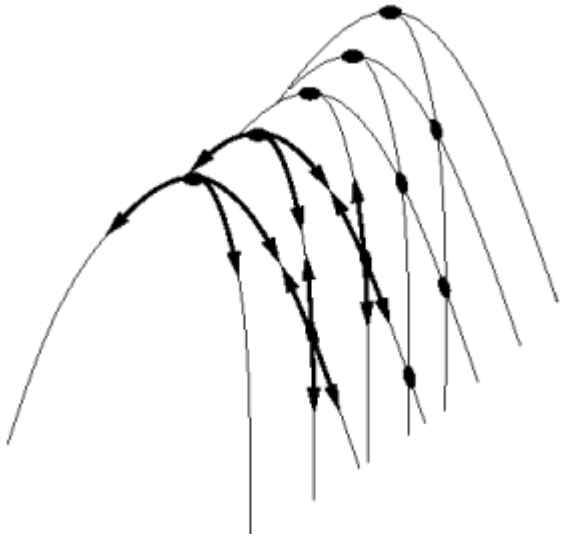
- melhor movimento c5: 1→5, h=5



- Faltam dois passos para concluir, finalize !

Busca de subida de encosta





- Máximos locais, platôs, planícies, cordilheiras.

Busca de subida de encosta

- Movimento lateral para evitar platôs
  - pode ocorrer repetição infinita se for um máximo local plano que não seja planície;
  - Impor um limite sobre o número de movimentos laterais;
- Subida de encosta estocástica
  - Escolha um movimento de subida de maneira aleatória com uma certa probabilidade.
- Subida de encosta com reinício aleatório
  - Conduz várias buscas subida de encosta a partir de diversos estados iniciais gerados ao acaso, visando encontrar um objetivo.
  - É completa, pois irá acabar gerando um estado objetivo como estado inicial. Porém ineficiente...

Busca de têmpera simulada (simulated annealing)

- Técnica inspirada no esfriamento de ligas metálicas para aumentar o seu grau de dureza.
- O processo começa em uma alta temperatura e prossegue com um resfriamento até atingir uma estabilidade.
- 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)

- Vamos mudar nosso ponto de vista de subida de encosta para descida de gradiente (isto é minimização de custo) e imaginar a tarefa de colocar uma bola de ping-pong na fenda mais profunda em uma superfície acidentada. Se deixarmos a bola rolar, ela acabará em um mínimo local.
- O truque é agitar apenas com força suficiente para fazer a bola sair dos mínimos locais, mas não o bastante para desalojá-la do mínimo global.
- A solução de têmpera simulada é começar a agitar com força (isto é em alta temperatura) e depois reduzir gradualmente a intensidade da agitação (ou seja, baixar a temperatura.)

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.

Algoritmos Genéticos

- Variante de Busca em Feixe Local estocástica.
- Os estados sucessores são gerados a partir da combinação de dois estados pais
- Analogia com seleção natural sexuada

Algoritmos Genéticos

- **População**: os k estados em cada geração.
- **Indivíduo**: cada estado representado como uma cadeia de caracteres (string)
- **Função de adaptação** (fitness): função de avaliação dos indivíduos que os classifica em um ranking. Valores mais altos para estados melhores.
- **Cruzamento** (crossing): troca de trechos de string entre dois indivíduos para gerar dois novos indivíduos.
- **Mutação**: troca aleatória de letras da string de um indivíduo

Produz a próxima geração de estados por seleção, mutação e crossover.

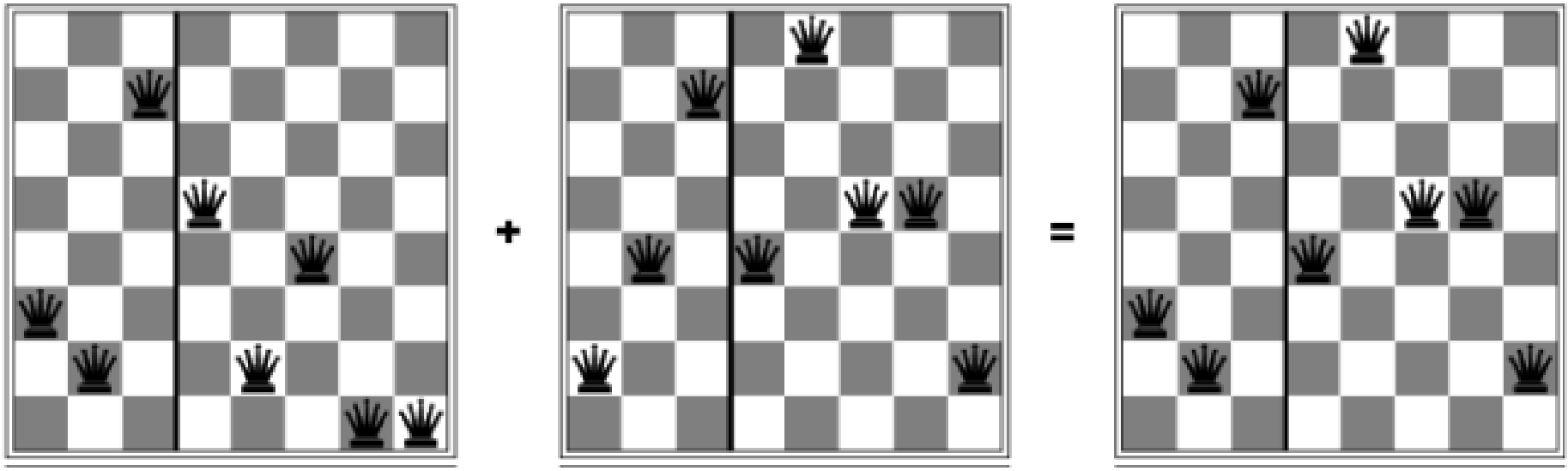
Algoritmos Genéticos

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

Algoritmos Genéticos

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



Algoritmos Genéticos

função ALGORITMO-GENETICO(população, FN-ADAPTA) **retorna** um indivíduo

Entrada: população (conjunto de indivíduos)

FN-ADAPTA (Função de adaptação do indivíduo)

Repetir

nova_população = conjunto vazio

Para a = 1 **até** TAMANHO(população) **faça**

x = SELEÇÃO-ALEATÓRIA(população, FN-ADAPTA)

y = SELEÇÃO-ALEATÓRIA(população, FN-ADAPTA)

filho = REPRODUZ(x, y)

Se (houver uma pequena probabilidade) **então** MUTAÇÃO(filho)

Adicionar filho a nova_população

população = nova_população

Até que alguns indivíduos sejam suficientemente adaptados ou passou um tempo suficiente

Retornar o melhor indivíduo na população segundo FN-ADAPTA

Algoritmos Genéticos

função REPRODUZ(x, y) **retorna** um indivíduo

Entrada: x, y (indivíduos pais)

n = COMPRIMENTO(x)

c = número aleatório entre 1 e n

filho = CONCATENA(SUBCADEIA(x, 1, c), SUBCADEIA(y, c+1, n))

Retornar filho

Algoritmos Genéticos

função REPRODUZ(x, y) **retorna** um indivíduo

Entrada: x, y (indivíduos pais)

n = COMPRIMENTO(x)

c = número aleatório entre 1 e n

filho = CONCATENA(SUBCADEIA(x, 1, c), SUBCADEIA(y, c+1, n))

Retornar filho