

Busca em Ambientes Complexos

Profa. Mercedes Gonzales
Márquez

Tópicos

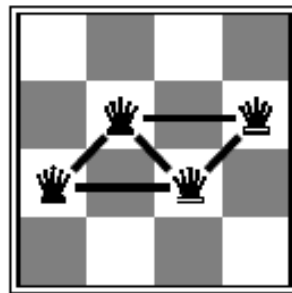
- Problemas de busca e otimização locais
 - Busca de subida de encosta
 - Têmpera Simulada
 - Busca em feixe local
 - Algoritmos Genéticos

Problemas de busca e otimização locais

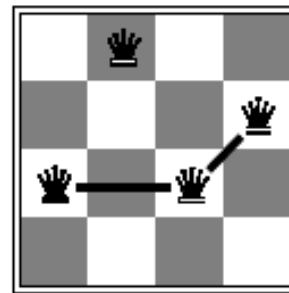
- Nos problemas de busca da aula anterior queríamos explorar caminhos pelo espaço de busca como o caminho de Curitiba a Fortaleza.
- Às vezes só nos importamos com o estado final, o caminho até o objetivo é irrelevante.
- Os algoritmos de busca local operam a partir de um estado inicial até os estados vizinhos, sem acompanhar os caminhos nem o conjunto de estados que foram alcançados.
- 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

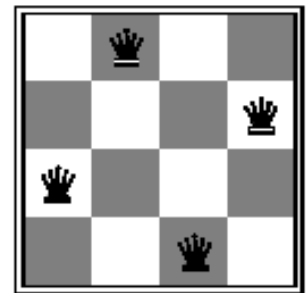
- Colocar n rainhas em um tabuleiro $n \times n$, sendo que cada linha, coluna ou diagonal pode ter apenas uma rainha, ou seja não deverá haver um par de rainhas se atacando uma a outra.



$h = 5$



$h = 2$



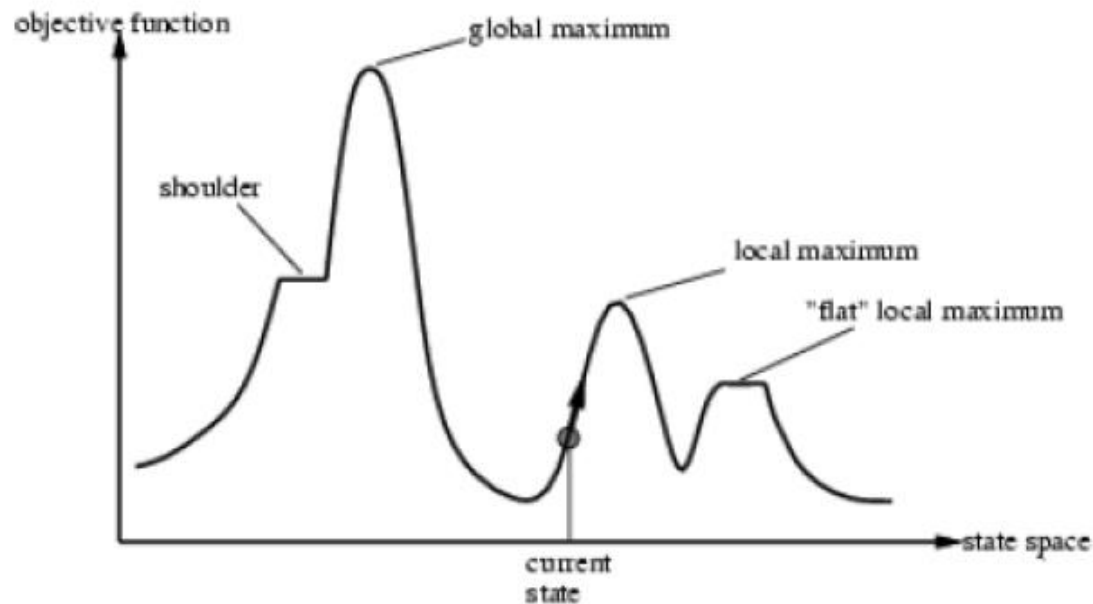
$h = 0$

- 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 a outra e só no último tabuleiro não temos nenhum par de rainhas atacando uma a outra.

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 íngreme**.
- 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.
- É como tentar alcançar o cume do Monte Everest em meio a um nevoeiro denso durante uma crise de amnésia.

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

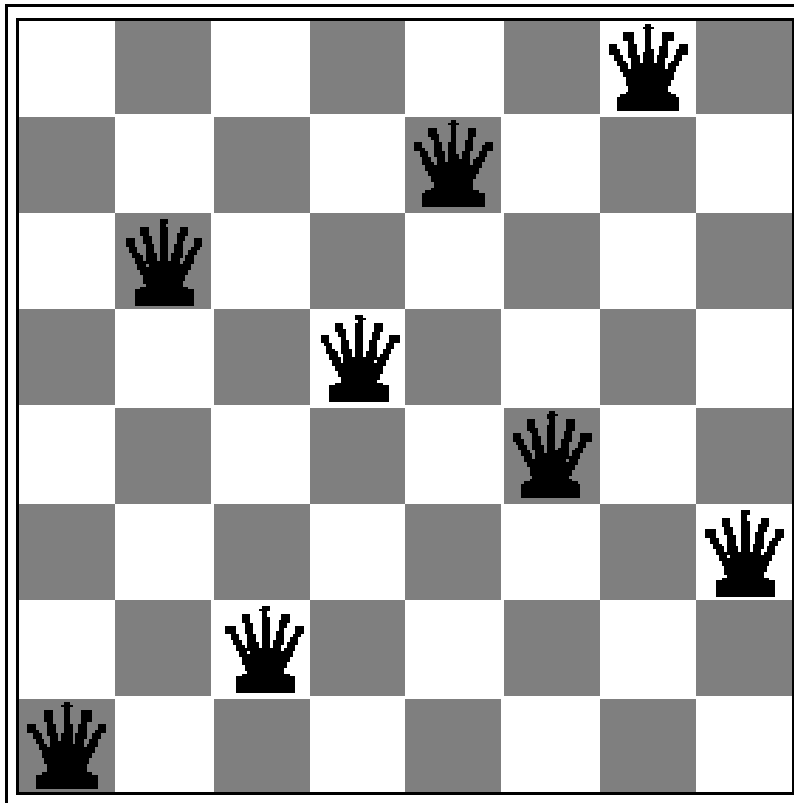
- Exemplo das 8 rainhas.
- Cada estado tem oito rainhas no tabuleiro, uma por coluna.
- Estado inicial escolhido aleatoriamente e
- Todos os sucessores de um estado são estados possíveis gerados pela movimentação de uma única rainha para outro quadrado na mesma coluna (de forma que cada estado tenha $8 \times 7 = 56$ 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

18	12	14	13	13	12	14	14
14	16	13	15	12	14	12	16
14	12	18	13	15	12	14	14
15	14	14		13	16	13	16
	14	17	15		14	16	16
17		16	18	15		15	
18	14		15	15	14		16
14	14	13	17	12	14	12	18

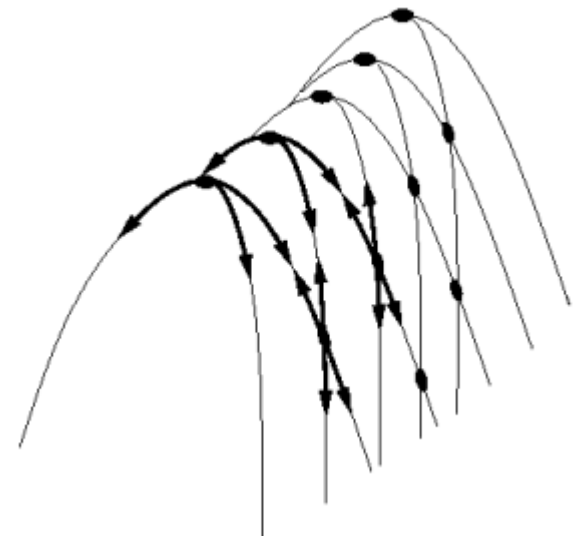
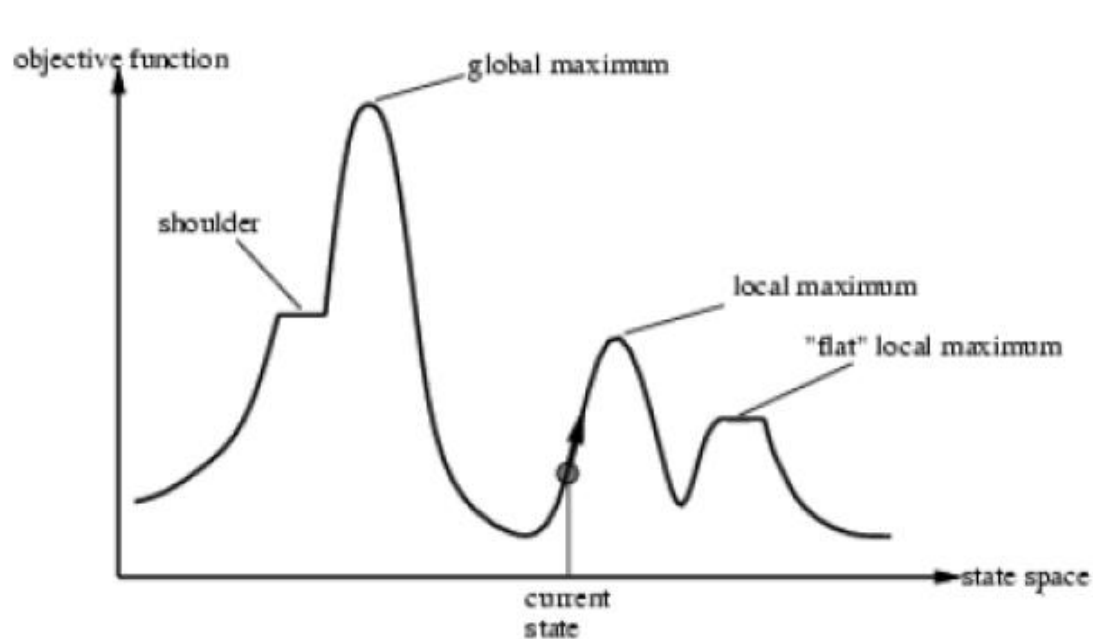
- $h=17$ para o estado ao lado.
- O tabuleiro mostra o valor de h para cada sucessor possível obtido pela movimentação de uma rainha dentro da sua coluna.
- Existem 8 movimentos que estão empatados como melhores, com $h=12$. O algoritmo de subida de encosta escolherá um deles.

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)

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

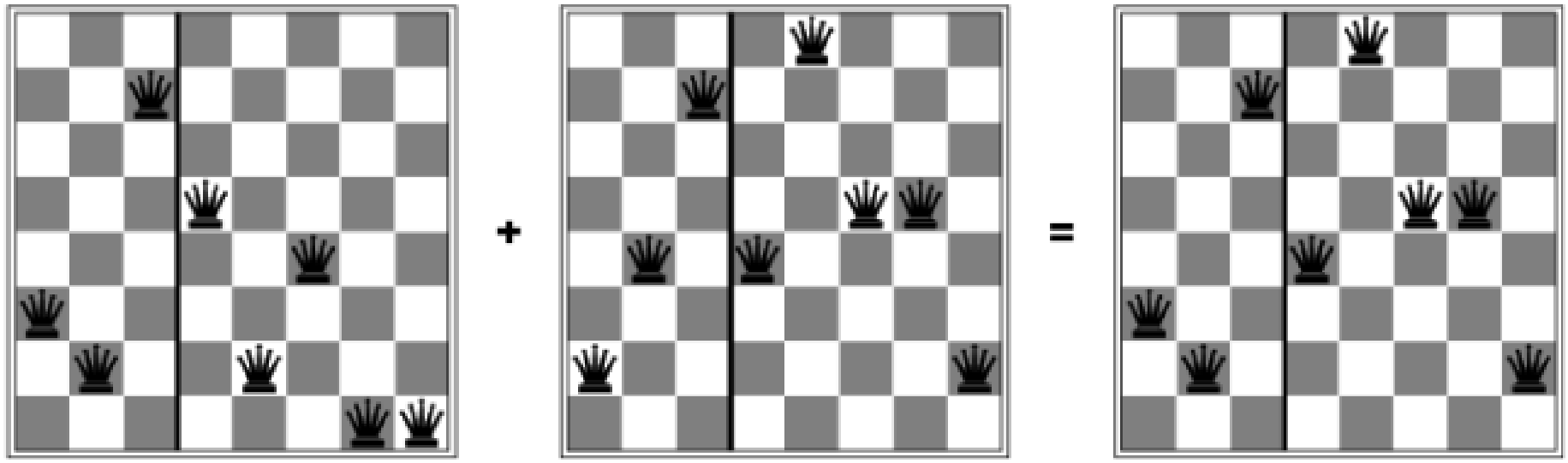
Problema das 8 rainhas



- Função de fitness: número de pares de rainhas que não estão se atacando ($\text{min}=0, \text{max}=8 \times 7/2=28$).
 $24/(24+23+20+11)=31$
 $23/(24+23+20+11)=29$, 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)

$n = \text{COMPRIMENTO}(x)$

$c = \text{número aleatório entre } 1 \text{ e } n$

$\text{filho} = \text{CONCATENA}(\text{SUBCADEIA}(x, 1, c), \text{SUBCADEIA}(y, c+1, n))$

Retornar filho

Algoritmos Genéticos

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

Entrada: x, y (indivíduos pais)

$n = \text{COMPRIMENTO}(x)$

$c = \text{número aleatório entre } 1 \text{ e } n$

$\text{filho} = \text{CONCATENA}(\text{SUBCADEIA}(x, 1, c), \text{SUBCADEIA}(y, c+1, n))$

Retornar filho