

Busca Local: Algoritmo *Hill Climbing*

Danilo Sipoli Sanches

Departamento Acadêmico de Computação
Universidade Tecnológica Federal do Paraná
Cornélio Procópio



Analogia

Metáfora

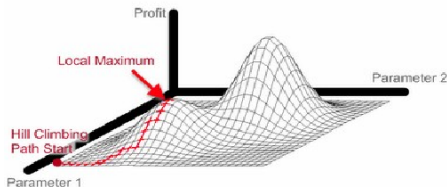
- Tentar encontrar o topo do Monte Everest sob uma forte neblina e sofrendo de amnésia



Hill Climbing

- Subida de encosta, subida de encosta mais íngreme, subida da colina, subida da montanha, **busca local gulosa**, ...
- Move de forma contínua em direção a valores crescentes da função objetivo (encosta acima)
- Examina apenas estados vizinhos imediatos ao corrente
 - não mantêm árvore de busca
 - armazena apenas estado corrente e tenta melhorá-lo
- Termina quando um pico (mínimo/máximo) é alcançado
 - nenhum vizinho tem um valor maior

The problem with hill climbing is that it gets stuck on "local-maxima"



Algoritmo

Função Hill-Climbing(Problema)

Início

EstadoAtual \leftarrow EstadoInicial(Problema)

Repita

Vizinho \leftarrow SucessorMaiorValor(EstadoAtual)

Se aval(Vizinho) \leq aval(EstadoAtual) **então**

retorna EstadoAtual

// não consegue encontrar algo melhor na vizinhança

EstadoAtual \leftarrow Vizinho

Fim_Repita

Fim

Hill Climbing - Exemplo

- **Problema:** 8 rainhas
- **Formulação:** estados completos
 - cada estado tem as 8 rainhas no tabuleiro, uma por coluna
- **Ações:** mover cada rainha para qualquer posição na mesma coluna
- **Função Sucessora:** dado um estado e uma ação, retorna o estado alcançado
 - para cada estado, há 56 estados sucessores: 8 rainhas x 7 posições
 - escolhe melhor sucessor corrente (aleatório, se existir vários)
- **Função de custo/heurística h:** número de pares de rainhas que se atacam
- **Estado objetivo:** configuração com $h=0$ (mínimo global)
 - somente para as soluções perfeitas (nenhuma rainha sendo atacada)

Hill Climbing - Exemplo

- **Problema:** 8 rainhas
- **Estado corrente:** $h=17$ (há 17 ataques entre rainhas)

Qual a melhor ação (por uma escolha gulosa)?

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

Hill Climbing - Exemplo

- **Problema:** 8 rainhas
- **Estado corrente:** $h=17$ (há 17 ataques entre rainhas)
- Em cada coluna é exibido o valor de h para cada possível sucessor

Qual a melhor ação (por uma escolha gulosa)?

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

$$(C_1 C_2 C_3 C_4 C_5 C_6 C_7 C_8) = (5 \ 6 \ 7 \ 4 \ 5 \ 6 \ 7 \ 6)$$

Hill Climbing - Exemplo

- Problema: 8 rainhas

Resposta:

Qualquer movimento que leve uma das rainhas na sua coluna a uma posição marcada com 12

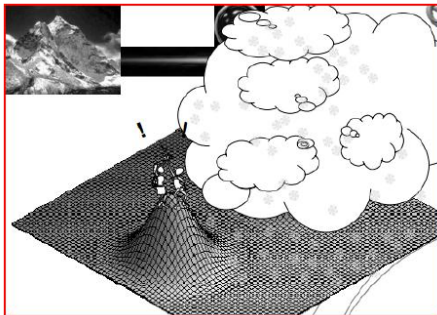
An 8x8 chessboard illustrating the 8-Queens problem. The board has alternating light and dark squares. Queens are placed on the board, and each square contains a numerical value representing a heuristic. Eight squares, each containing the number 12, are circled in red, indicating potential target positions for a queen move. A red line connects the queen in the first column (row 2) to the target square at (row 1, column 2).

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

- **Busca Gulosa Local** - escolhe sempre o primeiro melhor vizinho para progredir na busca
- Bons resultados em alguns problemas
- Capaz de progredir rapidamente para solução do problema

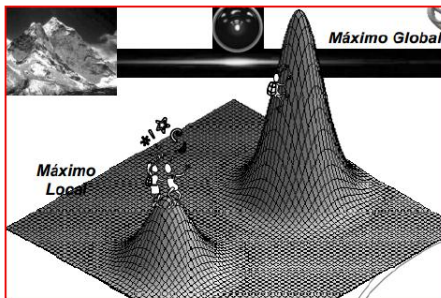
Problemas ?

Problemas ?



Hill Climbing - Problemas

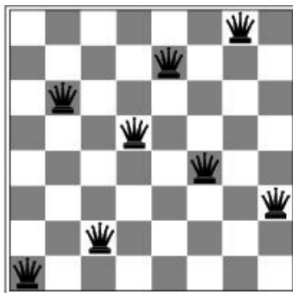
- Máximos locais



Hill Climbing - Exemplo

- **Problema:** 8 rainhas
- **Estado Corrente:** alcançada em 5 passos a partir do estado inicial

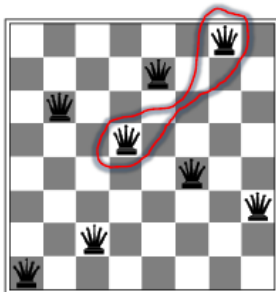
Mínimo Global?



Hill Climbing - Exemplo

- **Problema:** 8 rainhas
- **Estado Corrente:** alcançada em 5 passos a partir do estado inicial

Mínimo Local: todo sucessor tem custo mais alto



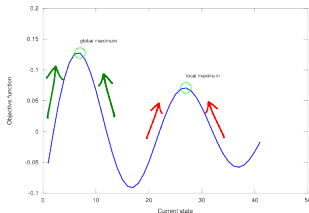
Um mínimo local com $h = 1$

Diferentes versões:

- **Original (maior passo):** move-se para o estado vizinho que proporciona a maior melhoria (redução ou aumento) no valor da função objetivo
- **First-Choice:** move-se para algum estado vizinho sorteado aleatoriamente e que esteja entre aqueles (se existirem) que melhoram o valor corrente da função objetivo
 - interrompe a busca após um limite pré-especificado de tentativas de movimento frustradas
 - interessante em problemas nos quais a quantidade de vizinhos é grande

Hill Climbing - Variações

- **Random-Restart:** múltiplas buscas hill climbing, a partir de diferentes estados iniciais gerados aleatoriamente
 - tenta evitar mínimos/máximos locais
 - interessante em problemas com superfícies de otimização complexas (muitos mínimos ou máximos locais não satisfatórios)
- Cada busca é executada
 - até que um número máximo estipulado de iterações seja atingido, ou
 - até que os resultados encontrados não apresentem melhora significativa
- Algoritmo escolhe o melhor resultado obtido com as diferentes buscas (diferentes reinícios)
- Cada execução produz apenas uma solução!



- Sucesso depende da topologia do espaço de estados
- Se houver poucos máximos locais e platôs
 - Hill climbing com reinício aleatório encontrará boa solução com rapidez
 - mesmo para mais complexos, pode encontrar máximo local razoavelmente bom
 - com um pequeno número de reinícios

- **Problema 1:** Hill Climbing nunca realiza movimentos “encosta abaixo” em direção a estados com valor mais baixo (ou de custo mais alto)
 - **incompleto** → porque pode ficar preso em um máximo local
- **Problema 2:** percurso aleatório, movendo para um sucessor escolhido ao acaso a partir do conjunto de sucessores
 - **completo** → extremamente ineficiente
- **Solução: ?**

- **Problema 1:** Hill Climbing nunca realiza movimentos “encosta abaixo” em direção a estados com valor mais baixo (ou de custo mais alto)
 - **incompleto** → porque pode ficar preso em um máximo local
- **Problema 2:** percurso aleatório, movendo para um sucessor escolhido ao acaso a partir do conjunto de sucessores
 - **completo** → extremamente ineficiente
- **Solução:** combinar Hill Climbing com um percurso aleatório que resulte de algum modo em eficiência e completeza
 - Simulated Annealing

Obrigado!