
Constraint Satisfaction Problems

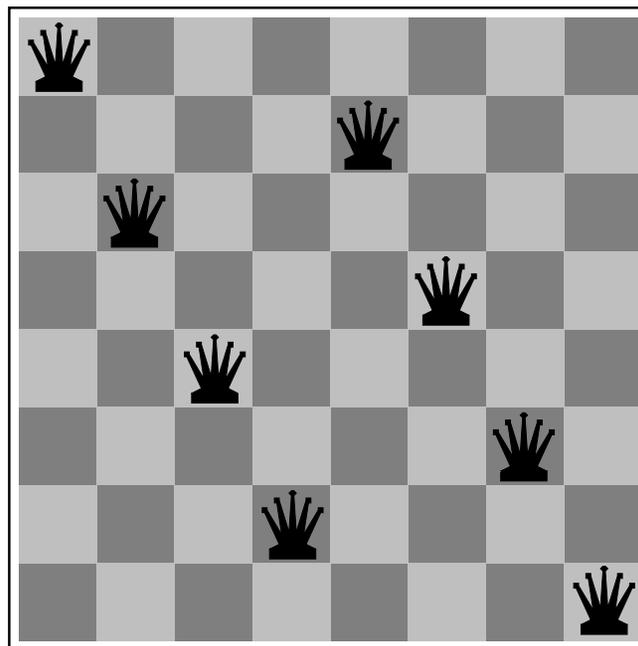
Fabrício Barth

Insper

Outubro de 2023

Problema das N rainhas

Dado um tabuleiro $N \times N$, é possível encontrar uma configuração onde N rainhas no tabuleiro não conseguem atacar nenhuma das outras rainhas no mesmo tabuleiro?



Introdução

Até agora vimos diversos problemas de planejamento. Problemas onde o agente precisava sair de um estado e chegar em outro, ou seja, sequenciar um conjunto de ações para alcançar um objetivo.

No entanto, existem outras classes de problemas onde o objetivo não está em encontrar uma sequência de ações, mas sim **encontrar um ou mais estados que satisfazem um conjunto de restrições.**

Algoritmo Subida da Montanha

Idéia: escolher sempre um sucessor melhor

(“*subir sempre*”).

function BSM-1(Estado *inicial*): Estado

Estado *atual* \leftarrow *inicial*

loop

prox \leftarrow melhor sucessor de *atual* (segundo *h*)

if $h(\textit{prox}) \geq h(\textit{atual})$ **then** {sem sucessor melhor}

return *atual*

end if

atual \leftarrow *prox*

end loop

Análise do algoritmo BSM

- Não mantém a árvore (logo, não pode retornar o caminho que usou para chegar à meta).
- Completo: **não** (problema de **máximos locais**)
- Ótimo: não se aplica
- Tempo: ?
- Espaço: **nada!**

Algoritmo Subida da Montanha Estocástico

```
function BSM-2(Estado inicial): Estado
Estado atual  $\leftarrow$  inicial
loop
    prox  $\leftarrow$  melhor sucessor de atual (segundo h)
    if  $h(\textit{prox}) \geq h(\textit{atual})$  then {sem sucessor melhor}
        if atual.éMeta() then
            return atual
        else
            atual  $\leftarrow$  estado gerado aleatoriamente
        end if
    else
        atual  $\leftarrow$  prox
    end if
end loop
```

Análise do algoritmo BSM Estocástico

- Completo: **sim** (se a probabilidade para geração de estados iniciais for equiprovável)
- Ótimo: não se aplica
- Tempo: ?
- Espaço: **nada!**

Material de **consulta**

- Capítulo 5 do livro do Russell & Norvig