

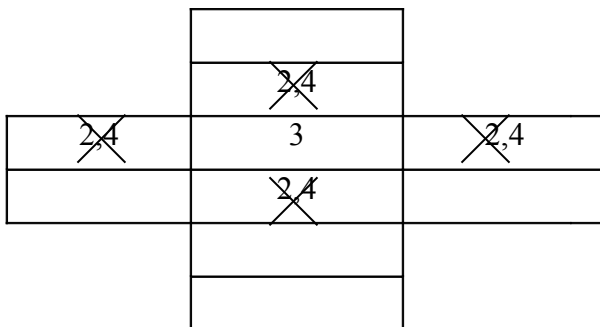


RESTRIÇÕES

Ex. 1) (Teste 14/7/07) Um quadrado latino de dimensão n é uma matriz $n \times n$ cujas entradas são números de 1 a n e em que não existem linhas nem colunas com números repetidos. Considere o problema de preencher uma matriz $n \times n$ formando um quadrado latino.

Formule este problema como um Problema de Satisfação de Restrições, indicando as variáveis, os domínios das variáveis e as restrições.

Ex. 2) (Teste 28/4/07) Considere a seguinte tabela, em que se deseja escrever os números de 0 a 9, sem repetições, de modo que nenhum número fique em contacto com o seu sucessor ou antecessor (por exemplo, se existir um 3 no quadrado central superior, não pode existir nem um 2 nem um 4 em acima, nem em abaixo, nem dos lados).



a) Formule este problema como um CSP, indicando as variáveis, os domínios das variáveis e as restrições.

b) Qual o conjunto das variáveis com que pode iniciar a procura em árvore considerando a heurística do maior grau? Justifique.

Ex. 3) Considere um quadrado mágico com a dimensão 3×3 . A cada posição é atribuído um valor inteiro de 1 a 9. (Todas as posições têm valores diferentes.) O objectivo é colocar os valores de tal forma que cada linha, coluna e diagonal (de comprimento 3) tenham a mesma soma. Segue-se um exemplo de um quadrado mágico:

4	3	8
9	5	1
2	7	6

Considere este problema como um CSP. Duas formulações distintas têm como variáveis e respectivos domínios:

a) Uma variável para cada posição, com o domínio [1..9], cujo valor corresponde ao inteiro contido nessa posição.

b) Uma variável para cada número, com domínio [1..9], cujo valor corresponde à posição onde se encontra o número.

Para qual das formulações é mais simples definir as restrições? Defina as restrições para a formulação mais simples.

Ex. 4) Considere o problema de cripto-aritmética que se segue:

$$\begin{array}{rcccccc} & & & E & A & T & \\ + & T & & H & A & T & \\ \hline A & P & P & L & E & & \end{array}$$

a) Formule este problema como um CSP. Sugestão: use como variáveis auxiliares X_1, X_2, X_3, X_4 , cujo valor é 0 ou 1, e que representam os dígitos das dezenas resultantes da adição.

b) Construa o respectivo grafo de restrições.

c) Qual o conjunto de variáveis com que pode iniciar a procura em árvore considerando a heurística do maior grau (“heuristic degree”)?

d) Considere que está a usar procura local juntamente com a heurística que minimiza o número de conflitos e que é devolvida a atribuição $A=4, E=8, H=2, L=3, P=0, T=9, X_1=1, X_2=0, X_3=1, X_4=1$. Quais as variáveis que o algoritmo vai considerar mudar? Suponha que o algoritmo decidiu aleatoriamente alterar o valor atribuído à variável A. Que valor deve ser atribuído a esta variável?

Ex. 5) Considere uma formalização do problema das 4-rainhas num CSP. As variáveis são x_1, x_2, x_3 e x_4 , todas elas com o domínio $\{1,2,3,4\}$. $x_j = i$ significa que existe uma rainha na posição (i,j) . Quais as restrições do problema? (Pode usar o operador $AllDiff(x_1, \dots, x_n)$, que obriga a que todos os valores das variáveis (x_1, \dots, x_n) tenham valores diferentes.)

Ex. 6) (Teste 22/7/06) Considere o problema do Sudoku para uma matriz 9×9 , com 9 sub-matrizes 3×3 . O objectivo é preencher todas as entradas com valores de 1 a 9, garantindo que cada número ocorre exactamente uma vez por linha, por coluna e por sub-matriz 3×3 . Formule este problema como um CSP, indicando as variáveis (e respectivos domínios) e as restrições. Pode usar o operador $AllDiff(x_1, \dots, x_n)$, que obriga a que todos os valores das variáveis (x_1, \dots, x_n) tenham valores diferentes.

Ex. 7) Considere um CSP com as seguintes características:

$$\begin{aligned} V &= \{x,y,z\} \\ D_x &= \{0,4,6,8\} \\ D_y &= \{3,6,9\} \\ D_z &= \{-1,1,2\} \end{aligned}$$

Restrições:

$$R1: \quad x + z = y$$

$$R2: 4x - y > 8$$

$$R3: 2x / z = 4$$

Realize as seguintes procuras usando ordem alfabética crescente para atribuir variáveis e ordem numérica crescente para atribuir valores.

- a) Procura com retrocesso sem efectuar qualquer tipo de inferência.
- b) Procura com retrocesso utilizando o algoritmo olhar-para-a-frente (*forward checking*).
- c) Procura com retrocesso utilizando consistência de arcos (de acordo com o algoritmo AC-3).