19.10.2022.

- concepte / noțiuni
- metode / clasificări
- tipuri de probleme / exerciții

Probleme de satisfacere a restricțiilor

Inteligență artificială

Cursul 4

Definirea acestui tip de probleme: 2 mulțimi: o stare: $\{X_i\} \to D$, mulțimea constrângerilor

Soluția → să dăm valori variabilelor a.î. toate restricțiile sunt satisfăcute

Acest tip de probleme sunt NP-hard, iar alg. s.n. scop-general, mai puternici decât backtracking

MaxCSP – maximizarea numărului de constrângeri satisfăcute

ex1) problema colorării unei hărți a.î. regiunile adiacente să fie colorate diferit.

pt harta Australiei avem variabilele corespunzătoare fiecărui teritoriu: WA, NT, Q, NSW, V, SA, T. (Australia)

Avem 3 culori = {red, green, blue} = D_i

Constrângerile = {regiunile adiacente trebuie să aibă culori diferite}

↓

WA ≠ NT sau (WA, NT) ∈ {(red, green), (red, blue), (green, red), (green, blue)... }

Soluția constă în atribuirea câte unei culori fiecărei variabile.

ex.2) problema celor N-regine

Var 1.

- Variabile: X_{ij}
- Domenii: {0,1}
  - 0 – nu avem o regină la poziția (i, j)
  - 1 – avem o regină

Constrângeri: $\sum_{ij} x_{ij} = N$

$\forall i,j,k, (x_{ij}, x_{ik}) \in \{(0,0),(0,1),(1,0)\}$

$\forall i,j,k, (x_{ij}, x_{kj}) \in \{(0,0),(0,1),(1,0)\}$

$\forall i,j,k \ (x_{ij}, x_{i+k,j+k}) \in \{(0,0),(0,1),(1,0)\}$

$\forall i,j,k \ (x_{ij}, x_{i+k,j-k}) \in \{(0,0),(0,1),(1,0)\}$

Varianta 2: avem variabilele care reprezintă liniile, iar valorile sunt coloanele.

Variabile: Q_k

Domenii: $\{1,2,3,\dots,N\}$.

Constrângeri: $\forall i,j$ non-threatening (Q_i, Q_j) (Implicit)

Explicit $(Q_1, Q_2) \in \{(1,3),(1,4)\dots\}$

ex 3. Sudoku

Variabile $= \{1,2,\dots,9\} \Rightarrow x_{ij}$

$\{1,2,\dots,9\} = N$

Restricții: toate valorile de pe o linie, coloană, regiune sunt diferite

$x_{ij} \neq x_{ik}, \forall k \neq i, j \in N$, (linie)

$x_{ij} \neq x_{kj}, \forall k \neq i, j \in N$ (coloana)

$x_{i_1 j_1} \neq x_{i_2 j_2} \ \forall (i_1, j_1) \neq (i_2, j_2) \in C_{ij}, \forall i,j \in N' = \{1,2,3\}$ (regiune), $\forall i,j \in N$

$C_{ij} = \{3(i-1)+i', 3(j-1)+j' \mid (i',j') \in N' \times N'\}$.

Există solvere care pot rezolva atât probleme de complexitate liniară, cât și probleme mai dificile.

ex.4 **Criptaritmetic puzzle:**

Variabile: F T U W R O X_1 X_2 X_3

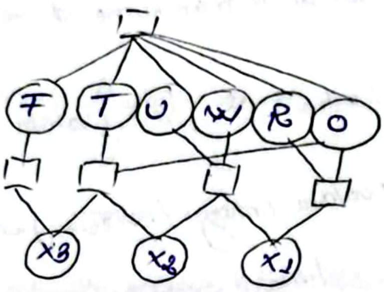
Domenii: {0,1,2,..., 9}.

Constrângeri (legat de sume): alldif (F, T, U, W, R, O)

$O + O = R + 10 \cdot X_1$.

Graf asociat problemei de satisfacere a restricțiilor / Factor Graph

$$\begin{array}{r} TWO \\ +\ TWO \\ \hline FOUR \end{array}$$



ex.5 **Problema planificării unor taskuri / Job-shop scheduling**

Construcția unei mașini a.î. că unele taskuri trebuie realizate înaintea altora (constrângere

$X = \{Axle_F, Axle_B, Wheel_{RF}, Wheel_{LF}, Wheel_{RB}, Wheel_{LB}, Nuts_{RF}, Nuts_{LF}, Nuts_{RB}, Nuts_{LB}, Cap_{RF}, Cap_{LF}, Cap_{RB}, Cap_{LB}, Inspect\}$.

ex. de restricții pt. acest exemplu.

1) Axele trebuie instalate înaintea roților $Axle_F + 10 \leq Wheel_{RF}$

2) După fixarea roților, strânge piulițele și apoi atașează capacele.

$Wheel_{RF} + 1 \leq Nuts_{RF}$; $Nuts_{RF} + 2 \leq Cap_{RF}$

3) Pt. a așeza axele se utilizează un instrument ($Axle_F$, $Axle_B$ nu se suprapun

$(Axle_F + 10 \leq Axle_B)$ or $(Axle_B + 10 \leq Axle_F)$

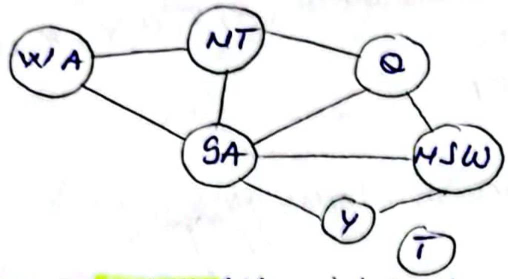
4) Inspecția este la sfârșit și durează 3 minute: $X + d_X \leq Inspect$.

5) Ansamblul trebuie terminat în 30 de minute: $D = \{1, 2, 3, \ldots, 27\}$

Graful de restricții → un graf asociat constrângerilor { nodurile → variabile
muchiile → constrângerile

3.8.

Pentru colorarea unei hărți graful va fi:



Variabile → discrete (domeniul de dimensiune d → complexitate exponențială $= d^m$, $m=$ nr. variabile)

ex. boolean CSPs, SAT (valori binare)

→ continue (valori întregi / caractere) ⇒ complexitate polinomială cu constrângeri liniare, care se pot rezolva prin met liniare.

Constrângeri → unare: o variabilă trebuie să ia o anumită valoare
→ binare
→ de ordin superior: implică cel puțin 3 variabile.
→ preferințe / restricții soft

Alg. de căutare.

O stare = o asignare (lucrăm cu asignări parțiale, adică nu toate variabilele iau valori)

{ Starea inițială: asignarea vidă, ∅
Funcția succesor: atribuie o valoare unei variabile neasignate
Testarea obiectivului: asignarea curentă este completă.

Alg. DFS se pot aplica pe acest tip de probleme, dar au complexitatea prea mare ⇒ folosim alg. de backtracking (obținut din DFS + CSP asignări)

Pt. a calcula complexitatea lui DFS trebuie să știm: factorul de ramificare (nd); adâncimea

Complexitatea pt. backtracking = d^n (n = nr. variabilelor
d = nr. nodurilor pe nivel
d^n = nr. frunzelor)

Căutarea backtracking este alg. de bază neinformat pt. CSP.

Met 1. Minimum remaining-values (MRV) = alg. la pasul curent, variabila care are cele mai puține valori permise.

Se fac aceste alegeri pt că este o euristică de tipul „fail-first", adică dacă vrem să greșim, să o facem cât mai devreme

Met 2 Least-constraining-value = alege valoarea cea mai puțin constrânsă.

Met1 + Met2 => putem rezolva pb. celor N-regine.

Alg. Forward Checking (pt. identificarea erorilor) → actualizează domeniul variabilelor neasignate atunci când selectăm o valoare pt. o variabilă, elimină valoarea din domeniul variabilelor neasignate, conectate cu aceasta.

Pt. N-regine → verificăm domeniul variabilelor următoare pt. a nu intra în conflict cu asignarea curentă

Propagarea constrângerilor

Arc consistency : poate fi aplicat după fiecare atribuire din alg. backtracking
cea mai simplă formă de propagare; face ca fiecare arc să fie consistent.

Definiție: pt. ∀x găsim o valoare în Y a.î. Y este consistent

Utilizarea tehnicilor de propagare a constrângerilor implică și o creștere a timpului de execuție.

Dacă propagarea durează mai mult decât căutarea, atunci nu se merită.

Alg. Conflict - directed BackJumping = ca un backtracking, doar că ne putem întoarce mai mult de un pas.