Rezolvarea problemelor cu ajutorul metodelor de căutare

Obiective

Formularea problemelor ca probleme de optimizare şi identificare modalităţilor de rezolvare a lor. Specificarea, proiectarea şi implementarea metodelor de căutare neinformată și informată.

Aspecte teoretice

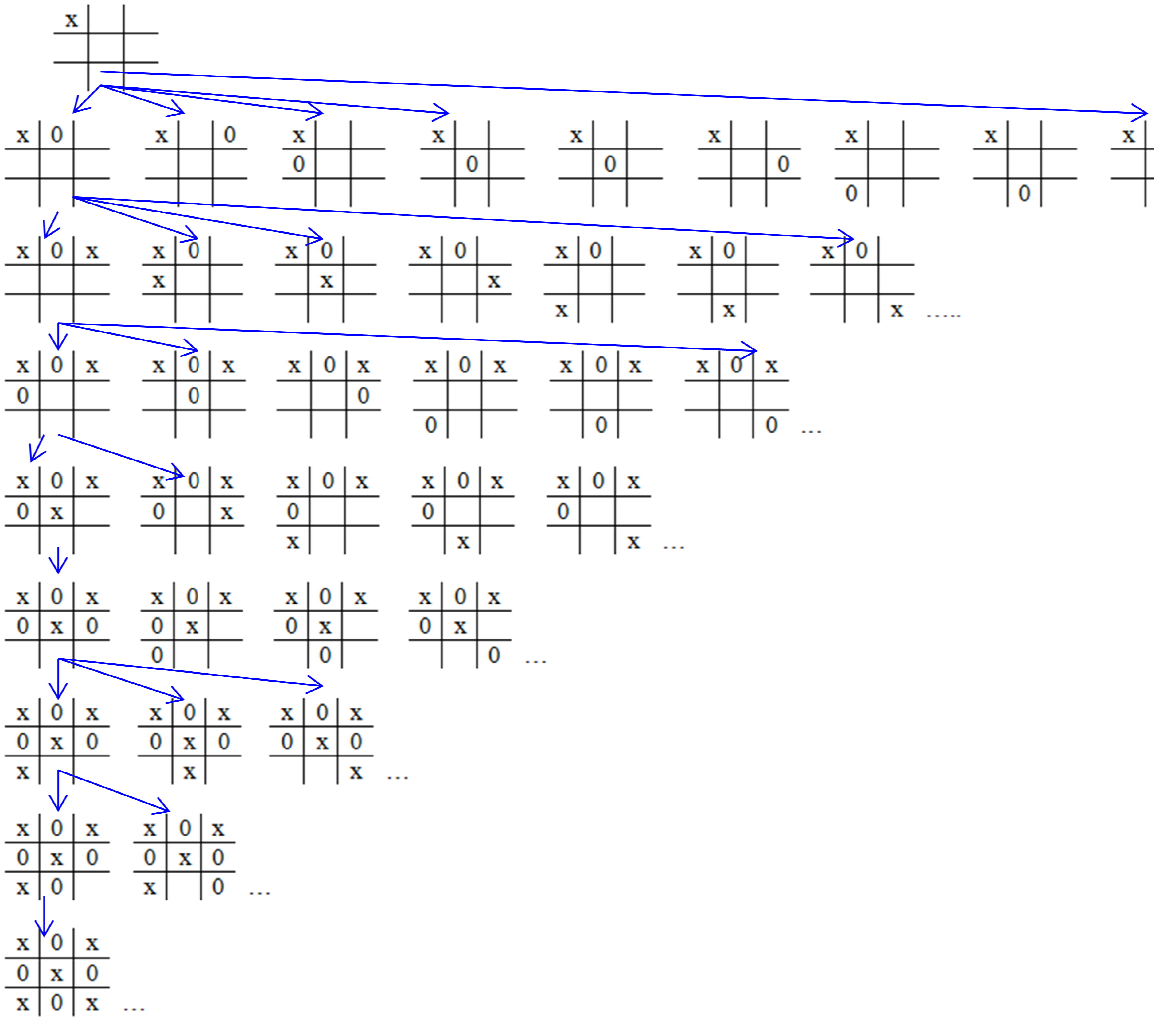
Rezolvarea problemelor ca proces de căutare
Tipuri de probleme de căutare.
Modalităţi de rezolvare a problemelor de căutare → Identificarea soluţiei potenţiale optime:

- Stabilirea componentelor problemei
    - Condiţii (constrângeri) pe care trebuie să le satisfacă (parţial sau total) soluţia
    - Funcţie de evaluare a unei soluţii potenţiale → identificareaa optimului
- Definirea spaţiul de căutare
- Stabilirea strategiei de identificare a soluţiei optime în spaţiul de căutare

Probleme abordate

1. Scurta prezentare a DFS, BFS, Greedy pe un arbore ”sintetic” cu adancime maxima 3
    a. ordinea de vizitare a nodurilor în cele 3 cazuri
    b. diferența majoră între DFS/BFS si Greedy (în cazul DFS/BFS nu avem costuri pe muchiile arborelui, în cazul Greedy avem costuri si putem prioritiza cumva vizitarea nodurilor)
2. Exemple de probleme - pentru fiecare problemă ar trebui discutate următoarele aspecte
    - cum modelăm stările spațiului de căutare
    - cum creăm arborele aferent spațiului de căutare (ce reprezentare folosim, dacă se reține tot arborele sau doar ”logica” lui)
    - în ce ordine vizităm nodurile

1. Exemple de aplicare a strategiilor de căutare neinformate în jocul Tic Tac Toe.
    - Crearea arborelui corespunzător spaţiului de căutare (doar primele nivele)
    - Parcurgerea arborelui conform strategiilor BFS, DFS



2. Rezolvarea problemei rucsacului.
   Enunţ: se dă un rucsac de capacitate M şi n obiecte, fiecare având o anumită greutate (w_i, i = 1,2,...,n) şi o anumită utilitate (u_i, i=1,2,...,n). Să se umple rucsacul cu obiecte astfel încât utilitatea obiectelor din rucsac să fie cât mai mare, iar greutatea obiectelor selectate să nu depăşească capacitatea rucsacului.

   BFS ...
   DFS ...
   Greddy: O posibilă euristică: alegem obiectele în ordinea descrescătoare a raportului u_i/w_i.

   Instanţa 1: M = 10, n = 3 – greedy găseşte soluţia

	A	B	C
w_i	8	4	6
u_i	3	4	6

   După ordonare C, B, A

   Alegem obiectul C ➔ $U_{obiecte_alese}$ = 6, $W_{obiecte_alese}$=6, loc gol = 4

Alegem obiectul B ➔ $U_{obiecte_alese}$ = 10, $W_{obiecte_alese}$=10, loc gol = 0

Instanţa 2: M = 5, n = 3 – greedy nu găseşte soluţia

	A	B	C
w_i	4	3	2
u_i	7	5	3

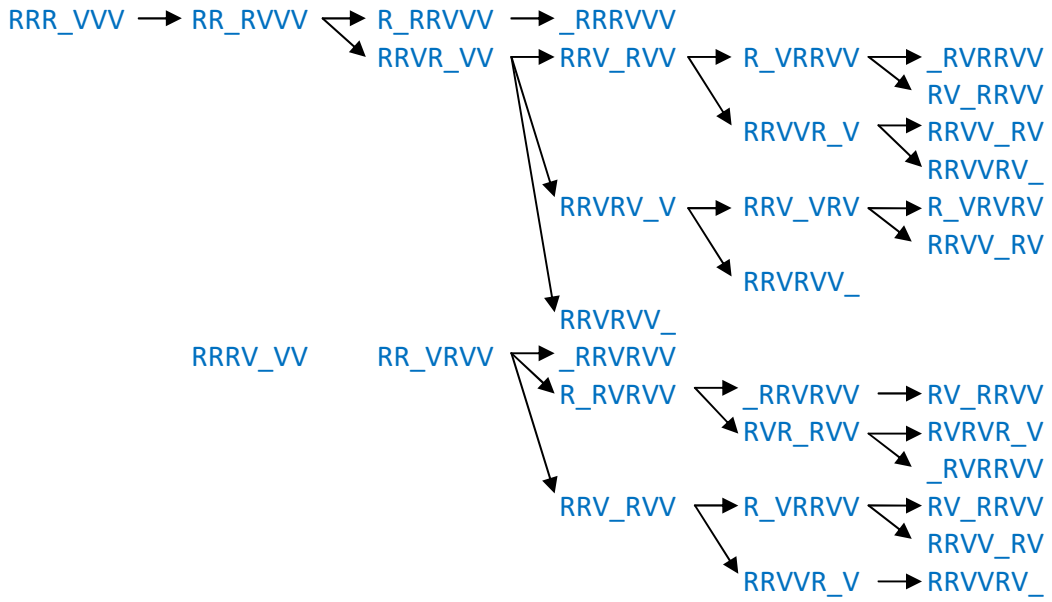
După ordonare A, B, C

Alegem obiectul A ➔ $U_{obiecte_alese}$ = 7, $W_{obiecte_alese}$=4, loc gol = 1 ➔ nu mai încap alte obiecte, dar soluţia optimă este dată de alegerea lui B şi C (U = 9, W = 5, loc gol = 0)

3. Problema comisului voiajor
   DFS
   BFS
   Greedy: o posibilă euristică e cel mai apropiat vecin.

4. Problema broscuţelor.
   Enunţ: 2 şiruri de broscuţe (un şir cu n broscuţe roşii şi un şir cu n broscuţe verzi) care se deplasează pe aceeaşi direcţie, dar în sensuri opuse, se întâlnesc la un moment dat. Şirurile de broscuţe se opresc astfel încât între ele rămâne un singur loc liber. Broscuţele vor să treaca unele în locul altora (cele roşii în locul celor verzi şi invers) dar pot efectua doar 2 tipuri de sărituri: „săritură simplă în faţă pe un loc gol” şi „săritură dublă în faţă peste o altă broscuţă (de aceeaşi culoare sau de culoare diferită) pe un loc gol”. Identificaţi succesiunea de mutări necesare inversării şirurilor de broscuţe.

   Instanţa 1: n = 3
   Configuraţia iniţială: RRR_VVV
   Configuraţia finală: VVV_RRR
   Spaţiul de căutare (incomplet) poate fi reprezentat astfel:



RRRVV_V → RRRVVV_

Euristici: f(n) = g(n)+h(n)
g(n) – nr de nivele parcurse în arborele de căutare până la nodul n
h(n) – nr de diferente între configuraţia broscuţelor fin nodul n şi configuraţia finală
de ex. Pt nodul n = RRV_RVV, g(n) = 3 h(n) = 5 (2 broscuţe – una verde şi una roşie – sunt deja la locul lor)