

## Rezolvarea problemelor cu ajutorul metodelor de căutare



### Obiective

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



### Aspecte teoretice

Rezolvarea problemelor ca proces de căutare

Tipuri de probleme de căutare.

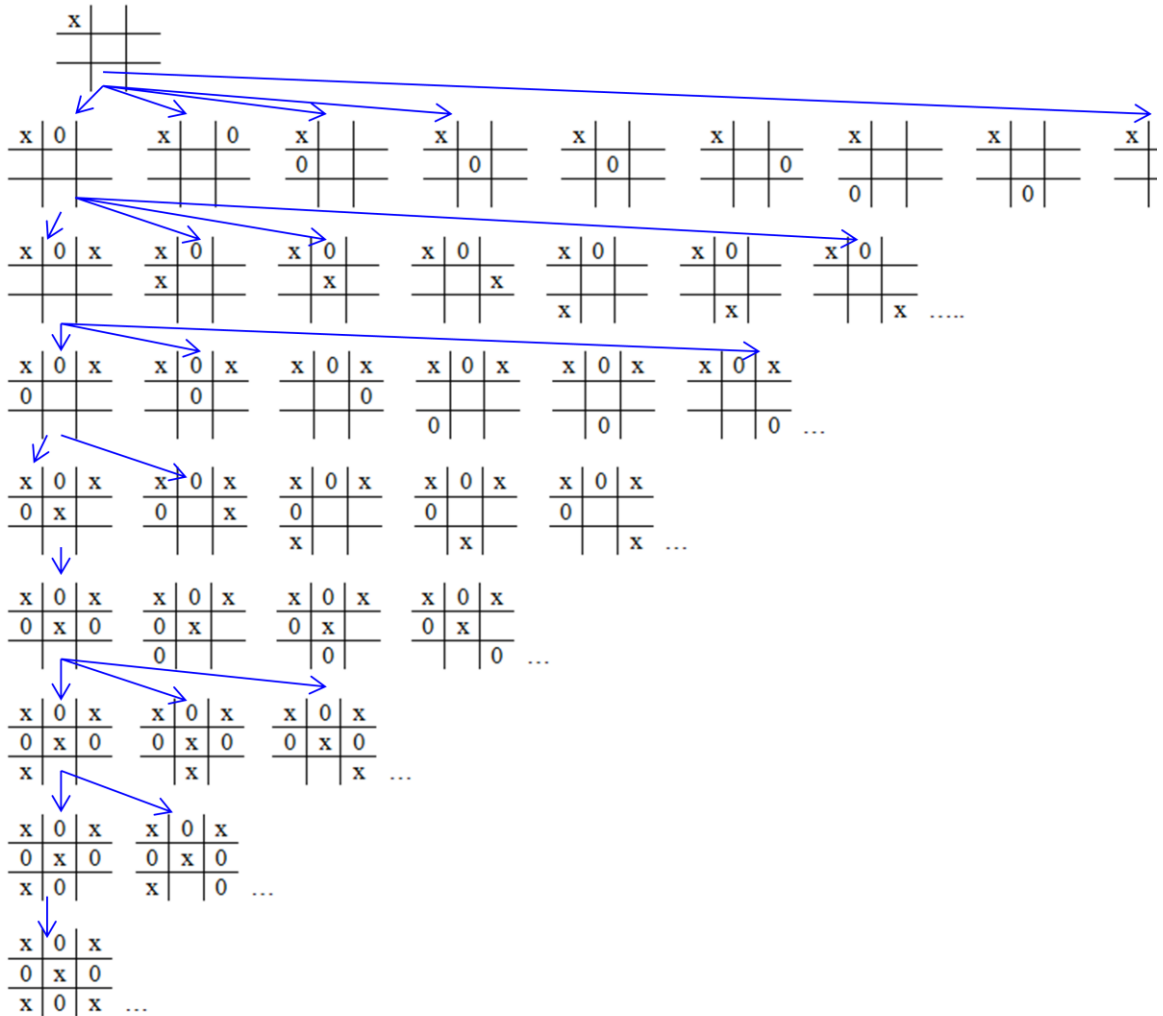
Modalități de rezolvare a problemelor de căutare → Identificarea soluției potențiale optime:

- Stabilirea componentelor problemei
  - Condiții (constrângeri) pe care trebuie să le satisfacă (parțial sau total) soluția
  - Funcție de evaluare a unei soluții potențiale → identificarea optimului
- Definirea spațiului de căutare
- Stabilirea strategiei de identificare a soluției optime în spațiul 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 și Greedy (în cazul DFS/BFS nu avem costuri pe muchiile arborelui, în cazul Greedy avem costuri și 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 spațiului de căutare (doar primele nivele)
  - Parcurgerea arborelui conform strategiilor BFS, DFS



2. Rezolvarea problemei rucsacului.

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

BFS ...

DFS ...

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

Instanța 1:  $M = 10, n = 3$  – greedy găsește soluția

	A	B	C
$w_i$	8	4	6
$u_i$	3	4	6

După ordonare C, B, A

Alegem obiectul C  $\rightarrow U_{\text{obiecte\_alese}} = 6, W_{\text{obiecte\_alese}}=6, \text{loc gol} = 4$

Alegem obiectul B →  $U_{\text{obiecte\_alese}} = 10, W_{\text{obiecte\_alese}}=10, \text{loc gol} = 0$

Instanța 2:  $M = 5, n = 3$  – greedy nu găsește soluția

	A	B	C
$w_i$	4	3	2
$u_i$	7	5	3

După ordonare A, B, C

Alegem obiectul A →  $U_{\text{obiecte\_alese}} = 7, W_{\text{obiecte\_alese}}=4, \text{loc gol} = 1$  → nu mai încap alte obiecte, dar soluția optimă este dată de alegerea lui B și C ( $U = 9, W = 5, \text{loc gol} = 0$ )

3. Problema comisului voiajor

DFS

BFS

Greedy: o posibilă euristică e cel mai apropiat vecin.

4. Problema broscuțelor.

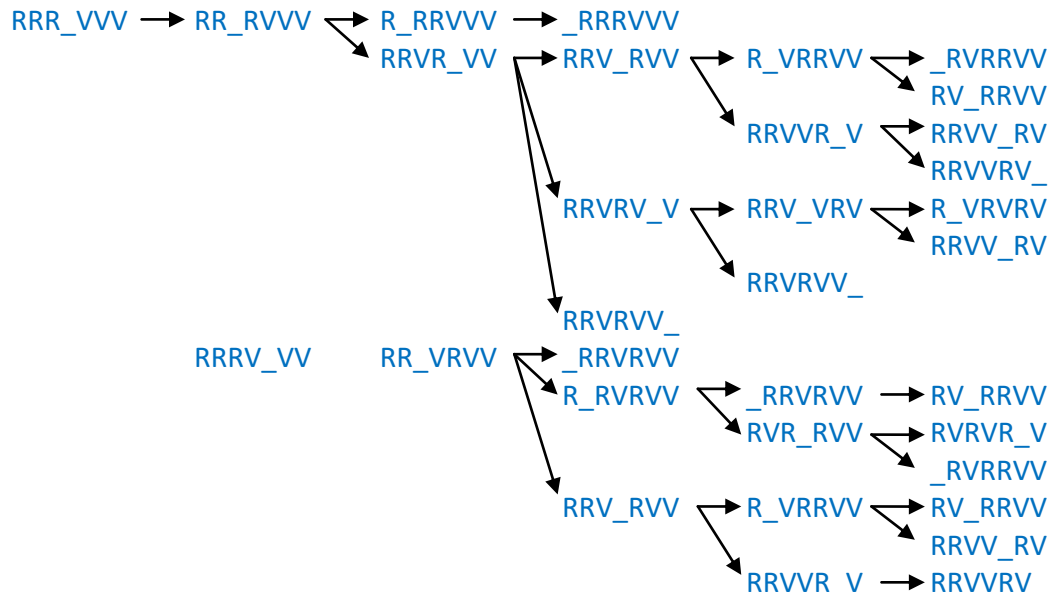
Enunț: 2 șiruri de broscuțe (un șir cu n broscuțe roșii și un șir cu n broscuțe verzi) care se deplasează pe aceeași direcție, dar în sensuri opuse, se întâlnesc la un moment dat. Șirurile de broscuțe se opresc astfel încât între ele rămâne un singur loc liber. Broscuțele vor să treacă unele în locul altora (cele roșii în locul celor verzi și invers) dar pot efectua doar 2 tipuri de sărituri: „săritură simplă în față pe un loc gol” și „săritură dublă în față peste o altă broscuță (de aceeași culoare sau de culoare diferită) pe un loc gol”. Identificați succesiunea de mutări necesare inversării șirurilor de broscuțe.

Instanța 1:  $n = 3$

Configurația inițială: RRR\_VVV

Configurația finală: VVV\_RRR

Spațiul 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 configurația broscuțelor fin nodul  $n$  și configurația finală

de ex. Pt nodul  $n = RRV\_RVV$ ,  $g(n) = 3$   $h(n) = 5$  (2 broscuțe – una verde și una roșie – sunt deja la locul lor)