

## Elemente de teoria jocurilor (I)

### Sinteză

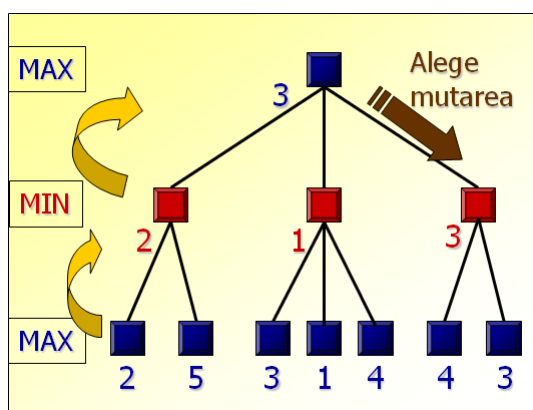
#### Jocuri secvențiale

*Algoritmul minimax* presupune căutarea mutării optime a calculatorului într-un arbore, în care nodurile sunt configurații ale tablei iar ramurile sunt mutările posibile. Se presupune că jocul este de sumă nulă, deci se poate folosi o singură funcție de evaluare pentru ambii jucători. Dacă  $f(n) > 0$ , poziția  $n$  este bună pentru calculator și rea pentru om, iar dacă  $f(n) < 0$ , poziția  $n$  este rea pentru calculator și bună pentru om. Funcția de evaluare este stabilită de proiectantul aplicației.

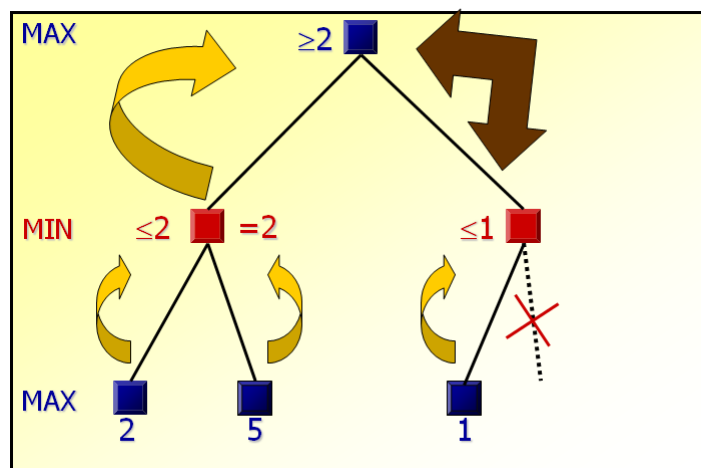
Pentru aplicarea algoritmului, se selectează o limită de adâncime și o funcție de evaluare. Se construiește arborele până la limita de adâncime. Se calculează funcția de evaluare pentru frunze și se propagă evaluarea în sus, selectând minimele pe nivelul minimizant (decizia omului) și maximele pe nivelul maximizant (decizia calculatorului).

*Retezarea alfa-beta* este o optimizare a algoritmului minimax, care întreține generarea arborelui cu propagarea valorilor. Unele valori obținute în arbore furnizează informații privind redundanța altor părți, care nu mai trebuie generate. În acest caz, se generează arborele în adâncime, de la stânga la dreapta și se propagă valorile finale ale nodurilor ca estimări inițiale pentru părinții lor. Valoarea din nodurile de pe un nivel maximizant este valoarea *alfa*, iar cea de pe un nivel minimizant este valoarea *beta*. De fiecare dată când  $\alpha \geq \beta$ , se oprește generarea nodurilor descendente.

În cazul cel mai favorabil, retezarea alfa-beta are complexitatea  $O(b^{d/2})$ , deci poate căuta pe o adâncime de două ori mai mare decât minimax.



*Minimax*



*Reducerea alfa-beta*

Pentru jocuri nedeterministe, se folosesc *valori așteptate* pentru nodurile-șansă, adică suma valorilor posibile înmulțite cu probabilitățile de apariție ale acestora.

*Căutarea pe arbori Monte Carlo* (engl. “Monte Carlo Tree Search”, MCTS) este o metodă stohastică de căutare a soluțiilor în jocuri cu factori mari de ramificare. Fiecare nod conține numărul de victorii / numărul de jocuri în care s-a trecut prin el. În prima fază, începând din rădăcină, se selectează în mod repetat acțiunea (nodul fiu)  $i$  care maximizează următoarea expresie:

$$\frac{w_i}{n_i} + \sqrt{\frac{2 \ln n}{n_i}}$$

unde:  $w_i$  este numărul de victorii în fiul  $i$ ,  $n_i$  este numărul de simulări în fiul  $i$ , iar  $n$  este numărul de simulări în nodul curent. Când nu se mai poate aplica selecția, urmează faza de expandare, în care este selectat în mod aleatoriu un nod fiu încă nevizitat și i se adaugă o nouă înregistrare de statistici (inițial 0/0). Începe apoi simularea Monte Carlo, în care se aleg mutări în mod aleatoriu până se ajunge într-o stare terminală (de exemplu, victorie sau înfrângere). După terminarea simulării, pentru toate pozițiile vizitate se incrementează numărul de jocuri și, dacă e cazul, numărul de victorii. După aplicarea algoritmului (în mod repetat), se alege mutarea cu cel mai mare număr de vizite  $n_i$ , deoarece valoarea sa este cel mai bine estimată.

### *Jocuri strategice*

*Reprezentarea unui joc în formă normală*, de exemplu, dilema deținutului (engl. “prisoner’s dilemma”):

		Deținutul 2	
		<i>Neagă</i>	<i>Mărturisește</i>
Deținutul 1	<i>Neagă</i>	-1, -1	-5, 0
	<i>Mărturisește</i>	0, -5	-3, -3

Fiecărei combinație de acțiuni îi corespund *câștiguri* (engl. “payoffs”) pentru cei doi agenți. De exemplu, dacă agentul 1 mărturisește și agentul 2 neagă, agentul 1 primește 0 iar agentul 2 primește -5. Agenții decid simultan, fără să știe unul decizia celuilalt. Fiecare agent trebuie să aleagă acțiunea optimă, care îi aduce un câștig maxim, indiferent de decizia celuilalt. Combinația acestor acțiuni optime corespunde unei stări de echilibru și este soluția jocului.

### *Dominanță*

O strategie  $S$  *domină* o strategie  $T$  dacă orice rezultat al lui  $S$  este cel puțin la fel de bun ca rezultatul corespunzător al lui  $T$ .

Dacă fiecare agent are o strategie dominantă și o joacă, atunci combinația acestora și câștigurile corespunzătoare constituie echilibrul strategiilor dominante ale jocului

*Principiul dominanței de ordin superior*: se elimină în mod iterativ strategiile dominate și se rezolvă jocul care rezultă în final, de exemplu:

	<i>C</i>	<i>D</i>	<i>E</i>
<i>A</i>	(1,1)	(2,0)	(3,-1)
<i>B</i>	(2,1)	(4,3)	(2,0)

### *Echilibru Nash pur*

O combinație de strategii este un *echilibru Nash* dacă fiecare agent își maximizează câștigul, date fiind strategiile folosite de ceilalți agenți. Echilibrul Nash identifică acele combinații de strategii care sunt *stabile*, în sensul că fiecare agent este mulțumit cu acțiunea aleasă, date fiind acțiunile celorlalți. Din stările de echilibru, niciun agent nu-și poate mări câștigul prin schimbarea *unilaterală* a strategiei.

*Calculul echilibrelor Nash pure.* Se evidențiază maximele pe linii pentru primul agent cu „{” . Se evidențiază maximele pe coloane pentru al doilea agent cu „}” . Stările încadrate de „{ }” sunt echilibre Nash pure.

		Deținutul 2	
		Neagă	Mărturisește
Deținutul 1	Neagă	-1, -1	-5, 0 }
	Mărturisește	{ 0, -5	{ -3, -3 }