

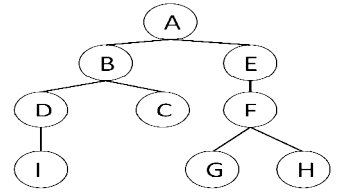
Subiectul 3

1. (1p) Precizați ordinea în care algoritmul de căutare în lățime (BFS) poate vizita nodurile arborelui de mai jos.

a. ABCDEFGHI b. ABDICEFGH c. ABEDCFIGH

Răspuns: c

2. (1p) Descrieți și exemplificați operatorul de încrucișare intermediară folosit de algoritmi evolutivi.



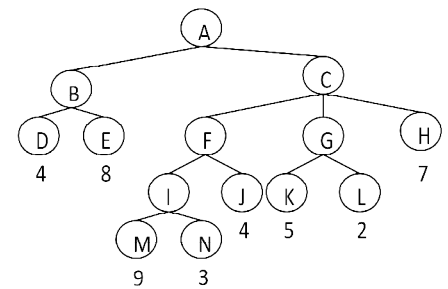
Răspuns: op pentru reprezentare reala. Gena descendentului va fi o combinație (liniară, convexă, geometrică, etc) a genelor din părinți. Poate fi singulară, simplă sau completă. Vezi exemplu din curs ☺

3. (1p) Precizați o diferență între algoritmi evolutivi generaționali și cei de tip steady-state.

Răspuns: descendenții obținuți prin op genetici (sel, xo și mutație) sunt adăugați într-o generație nouă (în cazul algoritmilor generaționali), respectiv înlocuiesc anumiți cromozomi (cei mai slabi) din generația curentă (în cazul alg steady-state)

4. (1p) Se dă arborele de joc de mai jos pentru un joc cu 2 jucători. Fiecare frunză este etichetată cu un număr întreg reprezentând scorul acordat acelei stări. Să se folosească algoritmul MiniMax pentru a eticheta toate nodurile arborelui.

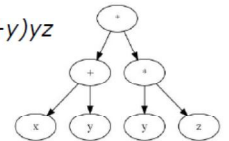
Răspuns: I - 3, F - 4, G - 5, B 4, C -4, A -4



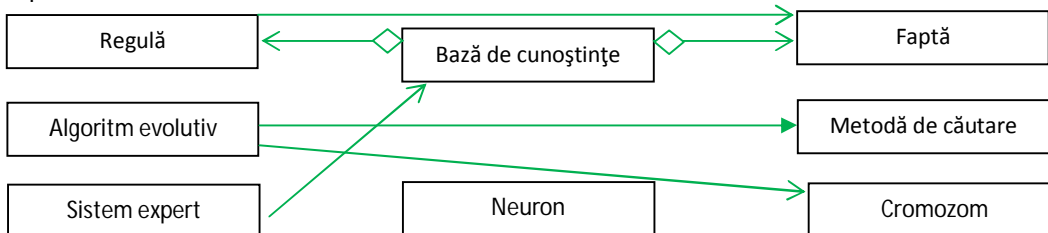
5. (1p) Prezentați un cromozom folosit în Programarea Genetică (reprezentare și fitness) pentru o problemă de regresie.

Răspuns: Pp o problemă de regresie cu 3 atribute (x_i, y_i, z_i). o singură ieșire reală (w_i) și $n=3$ exemple ($i=1,2,3$).

Un cromozom de GP este un arbore care codează o expresie matematică validă (corectă) sub forma unei funcții care depinde de x, y, z și anumiți coeficienți (constante). Nodurile arborelui pot conține astfel funcții (operatori, precum $+$, $-$, $*$, \sin , \cos , ...) sau argumente (x, y, z , constante). Pentru a calcula fitnessul unui cromozom se evaluează expresia codată de acel cromozom pentru fiecare exemplu, se compară rezultatul obținut cu valorile dorite (w) și se însumează diferențele absolute între ceea ce trebuia calculat și ceea ce se calculează pentru toate exemplele (se însumează abaterile valorilor calculate de la valorile țintă). Un exemplu de cromozom este prezentat în figură.



6. (1p) Precizați legăturile (și tipul lor – utilizare, generalizare, implementare, dependență, etc) existente între următoarele concepte:



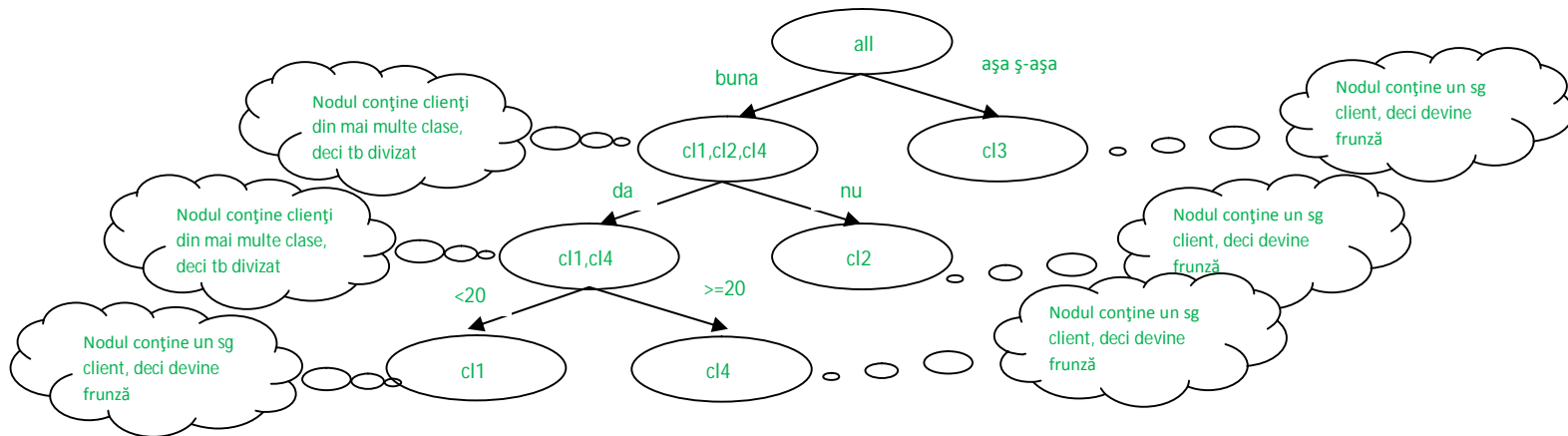
7. (1p) Se dau următoarele informații despre clienții unui restaurant: cum apreciază ei calitatea mâncării, dacă a fost aglomerat sau nu restaurantul, prețul plătit pentru o masă, dacă vor reveni sau nu. Să se construiască un arbore de decizie pentru o astfel de bază de cunoștințe.

Nr crt	Calitate mâncare	Aglomerare	Preț	Revin sau nu?
1	Bună	Da	15	Nu
2	Bună	Nu	20	Da
3	Așa ș-așa	Da	12	Nu
4	Bună	Da	25	Da

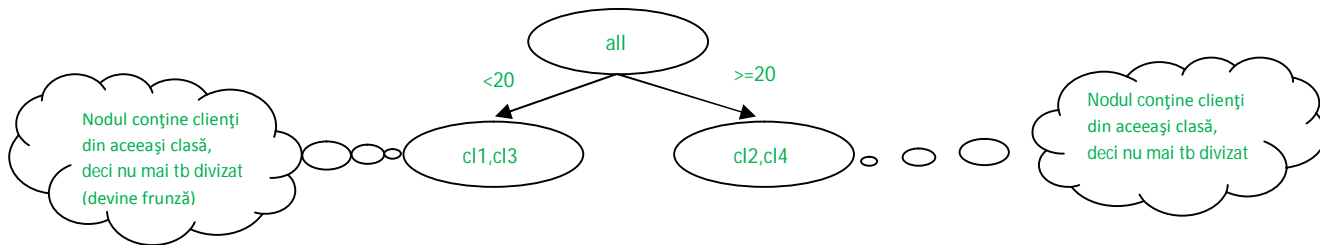
Răspuns: Este posibilă construirea mai multor arbori în funcție de ordinea în care se consider atributele. Ordinea poate fi dată de sondaj (ordinea coloanelor) sau anumite preferințe sau măsuri statistice (ca de ex câștigul

de informație – pt formule vezi cursul)

Dacă se consideră atributele calitate, aglomerație, preț se obține arborele:



Dacă se consideră atributele preț, calitate, aglomerație se obține arborele:



Sau alte cazuri...

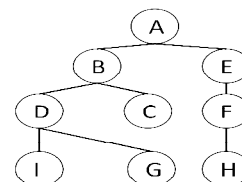
8. (2p) Se dă următoarea problemă: *Au fost înregistrate date privind nivelul temperaturii de-a lungul unei perioade date și se dorește previziunea acestei temperaturi pentru o dată ulterioară.* Să se propună o metodă de rezolvare (categoria de probleme din care face parte, algoritmul inteligent și ideea de bază a lui, evaluarea performanței algoritmului).

Răspuns: problemă de regresie, care poate fi rezolvată cu un algoritm de învățare supervizată (de exemplu o RNA). Pentru a rezolva problema cu RNA, stabilim că datele de intrare sunt date sub forma perechilor ($momTimp_i$, $valTemp_i$) unde $momTimp_i$ reprezintă data de intrare (de tip număr întreg pozitiv) și $valTemp_i$ data de ieșire (de tip număr real), iar $i=1,2,3,\dots,n$ (n nr de înregistrări). Deci, va fi nevoie de o RNA cu un sg nod pe stratul de intrare, un sg nod pe stratul de ieșire și un anumit nr de straturi ascunse (să pp că avem 1 strat ascuns cu 10 neuroni). Neuronii de pe straturi consecutive sunt conectați prin legături ponderate w_{ij}^k (ponderea conexiunii între al i -lea neuron de pe stratul $k-1$ și al j -lea neuron de pe stratul k). Fiecare neuron are o funcție de activare (să pp funcția liniară – vezi în curs formula). Ideea de bază după care lucrează RNA: plecând de la valori oarecare ale ponderilor încearcă să găsească valorile optime ale acestora astfel: se activează RNA, se stabilește eroarea de calcul (ca diferență absolută între ceea ce trebuia calculat, $valTemp_1$, și ceea ce s-a calculat) și se propagă eroarea înapoi în rețea, modificându-se ponderile conexiunilor. Se consider următorul exemplu ($momTimp_2$, $valTemp_2$) și tot așa până se ajunge la un nr de epoci (o epocă constând în considerarea tuturor exemplelor) sau l o eroare mai mică decât epsilon. Detalii despre activare, eroare și propagarea ei în curs și seminar

Subiectul 4

- (1p) Precizați ordinea în care algoritmul de căutare în adâncime (DFS) poate vizita nodurile arborelui de mai jos.
a. ABEDCFIGH b. ABDIGCEFH c. AEFHBCDGI

Răspuns: b,c



- (1p) Descrieți și exemplificați operatorul de mutație prin interschimbare folosit de algoritmi evolutivi.

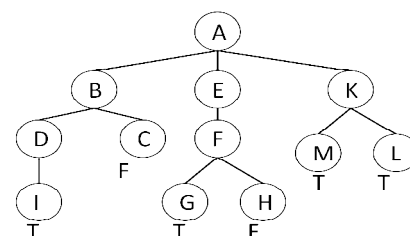
Răspuns: op pentru reprezentarea prin permutare care interschimbă valorile a 2 gene. De ex, pt cromozomul cu 5 gene 1 2 3 4 5, se aleg aleator pozițiile a 2 gene (poz1 = 2, poz2 = 4) și se interschimbă valorile acestor gene: 1 4 3 2 5

- (1p) Precizați o diferență între operatorul de încrucișare intermediară simplă și cel de încrucișare intermediară completă folosiți de algoritmi evolutivi.

Răspuns: o diferență constă în numărul de gene ale părinților considerate pentru încrucișare: op de încrucișare intermediară simplă amestecă toate genele de după o anumită poziție (de la acea poziție până la sfârșit) din părinți, iar cel de încrucișare intermediară completă amestecă toate genele din părinți.

- (1p) Se dau arborele de joc de mai jos pentru un joc cu 2 jucători. Fiecare frunză este etichetată cu True sau False reprezentând posibilitatea de câștig a jucătorului care trebuie să mute. Să se folosească algoritmul AndOr pentru a eticheta nodurile arborelui.

Răspuns: D – T, F – T, B – F, E – T, K – T, A – T

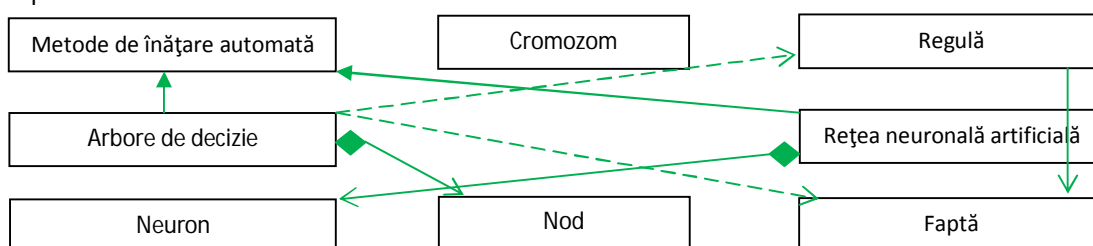


- (1p) Precizați etapele de realizare (și rolul lor) ale unui sistem bazat pe reguli (într-un mediu fuzzy).

Răspuns:

- Definirea intrărilor și ieșirilor de către expert (datele de intrare și ieșire brute, fuzzificarea datelor de intrare și ieșire prin stabilirea variabilelor fuzzy și a mulțimilor fuzzy pe baza funcțiilor de apartenență)
- Construirea unei baze de reguli de către expert (posibil sub forma unei matrici de decizie a bazei de cunoștințe)
- Evaluarea regulilor (adică inferența – transformarea intrărilor fuzzy în ieșiri fuzzy prin aplicarea regulilor din baza de cunoștințe)
- Agregarea rezultatelor (unificarea ieșirilor tuturor regulilor aplicate)
- Defuzificarea (transformarea rezultatului fuzzy al agregării într-o valoare crisp)
- Interpretarea rezultatelor de către expert

- (1p) Precizați legăturile (și tipul lor – utilizare, generalizare, implementare, dependență, etc) existente între următoarele concepte:

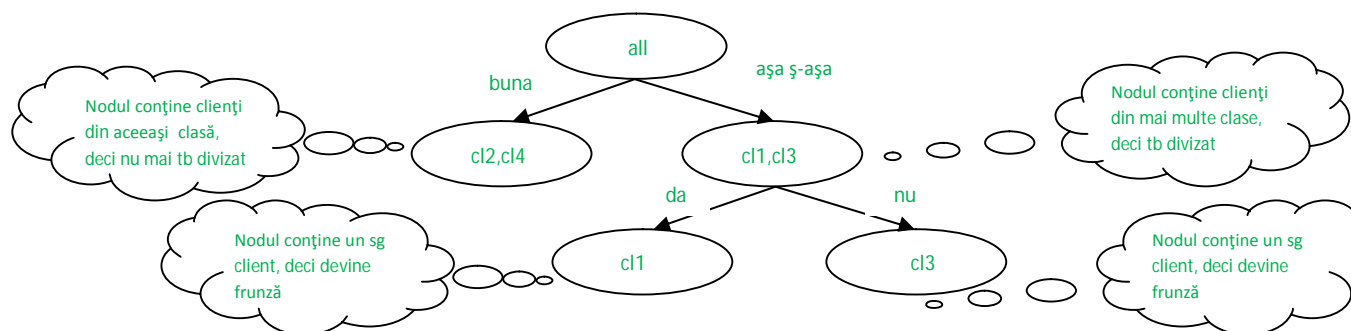


7. (1p) Se dau următoarele informații despre *clienții unui restaurant*: cum apreciază ei calitatea mâncării, dacă a fost aglomerat sau nu restaurantul, prețul plătit pentru o masă, dacă vor reveni sau nu. Să se construiască un arbore de decizie pentru o astfel de bază de cunoștințe.

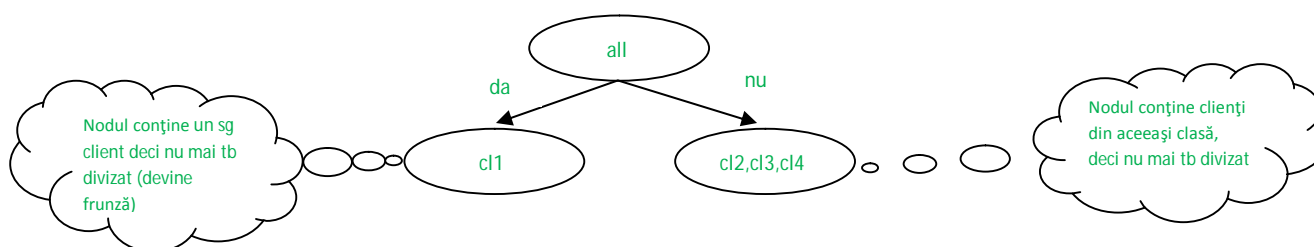
Nr crt	Calitate mâncare	Aglomerare	Preț	Revin sau nu?
1	Așa ș-așa	Da	15	Nu
2	Bună	Nu	20	Da
3	Așa ș-așa	Nu	12	Da
4	Bună	Nu	25	Da

Răspuns: Este posibilă construirea mai multor arbori în funcție de ordinea în care se consider atributele. Ordinea poate fi dată de sondaj (ordinea coloanelor) sau anumite preferințe sau măsuri statistice (ca de ex câștigul de informație – pt formule vezi cursul)

Dacă se consideră atributele calitate, aglomerație, preț se obține arborele:



Dacă se consideră atributele aglomerație, preț, calitate se obține arborele:



Sau alte cazuri...

8. (2p) Se dă următoarea problemă: un robot s-a tot plimbat printr-un labirint încercând să ajungă la marginea labirintului. El poate efectua deplasări în căsuțele vecine libere situate la dreapta, la stânga, mai jos sau mai sus față de el. Deplasarea într-o căsuță liberă îi aduce un câștig de 5 unități, iar deplasarea într-o căsuță de pe marginea labirintului îi aduce un câștig de 10 unități. Ajutați-l pe robot să învețe să ajungă la marginea labirintului. Să se propună o metodă de rezolvare (categoria de probleme din care face parte, algoritmul inteligent și ideea de bază a lui, evaluarea performanței algoritmului).

Răspuns: problemă de clasificare (robotul ajunge sau nu la marginea labirintului), care poate fi rezolvată cu un algoritm de învățare prin întărire (similar problemei din curs...) sau cu un algoritm de învățare supervizată (în care datele de intrare se dau sub forma drumurilor parcurse de robot și a succesului avut (a ieșit sau nu din labirint)).