

Lista temelor de proiect la disciplina

„Inteligență artificială aplicată”

pentru studenții anului 1 de la programul de master

„Medii virtuale multimodale și distribuite”,

semestrul 1

Condiții generale

Pentru fiecare proiect este necesar:

- De prezentat sumar ideea problemei
- de descris detaliat condițiile problemei,
- de formulat modelul matematic al problemei (dacă este cazul),
- de analizat (cel puțin să se menționeze) și alte moduri de soluționare a problemei din proiect,
- de a prezenta succint etapele de soluționare propuse,
- codul sursă al programului, care rezolvă problema din proiect, și explicațiile referitor la modul lui de organizare,
- proiectul trebuie prezentat cel târziu în penultima săptămână de studii în ziua de miercuri, dar așa încât să avem posibilitatea să-l putem analiza amical.
- *Ideea proiectului consta în comunicarea dintre agenți pentru a soluționa problema enunțată în proiect. Gândiți-vă că totul poate fi organizat ca o comunicare între niște agenți, unul dintre care dorește să rezolve o problemă (de exemplu problema propusă pentru proiect mai jos) și merge în “piață” și caută pe cineva poate să-l ajute, cu care *discută* condițiile problemei. Agentul care se apucă să rezolve problema poate să comunice la rândul său cu alți agenți pentru a rezolva probleme particulare suplimentare.*
- *Se accepta și proiecte realizate parțial, adică partea cu algoritmi (euristici) funcționează, partea cu agenți este relativ funcțională, doar sunt probleme la funcționarea lor ca un tot întreg.*
- *La implementarea soluțiilor algoritmice pentru problemele de mai jos se vor folosi algoritmi euristici, inclusiv algoritmi genetici, pe algoritmi de inteligență artificială distribuită.*

Observație. Nu lăsați totul pentru ultimele zile, căci va fi imposibil fizic să-i ascult și să analizez proiectele respective în penultima săptămână.

Lista proiectelor (incompletă, vor mai fi adăugate proiecte)

1. Să se modeleze modelul macroeconomic păienginiș (Cobb-Web Model) de funcționare a pieței libere pentru a stabili punctele de echilibru ale economiei.
2. Să se modeleze modelul microeconomic păienginiș (Cobb-Web Model) de funcționare a pieței libere pentru a stabili prețurile optime (de echilibru) ale produselor.
3. Se considera ca este necesar de a ne deplasa din orasul A în orasul B. Dar nu exista o ruta directă, sau ruta directă este cea mai costisitoare (de timp, de bani, de nervi, etc). Exista in sa o retea formata din x_1, \dots, x_N orase care au unele legaturi între ele. În fiecare din acele N orașe avem câte un prieten pe care ne propunem să-l vizităm o singură dată. Sa se determine ruta cea mai puțin costisitoare. // //
4. Consideram o variantă a problemei rucsacului. Fie avem un rucsac în care trebuie sa

punem unele obiecte. Forța noastră fiind limitată, nu putem duce mai mult de G kilograme. Și volumul rucsacului V este limitat. Este clar ca nu putem lua toate lucrurile cu noi. Atunci vom considera ca unele lucruri sunt mai importante/utile decât altele. Scopul este de a maximiza utilitatea sumară a obiectelor puse în rucsac. *(E o problemă în care avem 2 restricții netriviiale: volumul max $\leq V$, greutatea maximă $\leq G$. Putem folosi una din metodele de penalizare de rezolvare – (a)nivele fixe de penalizare pt fiecare restricție, (b)nivel dinamic de penalizare, (c)cea mai bună soluție nefeasabilă nu poate fi evaluată mai bine ca cea mai proastă soluție feasibilă, (d)examinarea succesivă a restricțiilor, (e)... --).*

5. Se dorește sa se aproximeze o funcție arbitrară $f(x)$ pe segmentul $[a, b]$ cu un polinom de gradul 5. Sa se determine coeficienții acelui polinom. Este clar că un rol important în soluționarea acestei probleme îl reprezintă sensul noțiunii aproximare a unei funcții de către altă funcție. // //
6. O familie dorește să-și aleagă locul de trai într-o localitate din țară. Fiecare are preferințele sale exprimate prin funcțiile f_1 și f_2 respectiv. Aceste funcții țin cont individual de fiecare dintre factori: x_1 – temperatura medie a mediului, x_2 – distanța de la casa părintească, x_3 – costul de trai, x_4 – dimensiunea orașului. Să se propună o soluție pentru problemă și să se argumenteze propunerea. // //
7. Se consideră o matrice rară, adică o matrice în care numărul de elemente diferite de zero este mult mai mic decât numărul total de elemente din matrice. Astfel de matrice par frecvent, de exemplu, în probleme de mecanică, care folosesc metoda elementelor finite pentru calculul unor parametri. Soluționarea sistemelor de ecuații liniare în care sunt implicate astfel de matrice în cazul când dimensiunile sistemului sunt mari reprezintă o dificultate relativă. O parte din aceste dificultăți pot fi eliminate dacă în matricea inițială liniile și coloanele sunt permutate astfel încât elementele diferite de zero ale matricei să fie situate de-a lungul diagonalei. Scopul proiectului: de a determina permutările respective astfel încât să se obțină o matrice maximal apropiată de matricea diagonală.// //
8. Se consideră o rețea neuronală, să zicem una multistrat de tip feed-forward. O rețea neuronală este perfectă pentru soluționarea unei probleme/categorii de probleme dacă ne dă răspunsul dorit fără eroare, sau eroarea este mică. Instruirea unei rețele neuronale presupune niște algoritmi de ajustare a ponderilor rețelei neuronale, care ne-ar conduce la ponderile optime. Să se elaboreze și să se folosească algoritmi genetici/euristici la antrenarea rețelei neuronale. ()
9. Să se elaboreze și să se implementeze un algoritm genetic pentru rezolvarea unei probleme de programare liniară. *(E o problemă în care funcția obiectiv este una liniară, și restricțiile sunt inegalități liniare. Putem folosi una din metodele de penalizare de rezolvare – (a)nivele fixe de penalizare pt fiecare restricție, (b)nivel dinamic de penalizare, (c)cea mai bună soluție nefeasabilă nu poate fi evaluată mai bine ca cea mai proastă soluție feasibilă, (d)examinarea succesivă a restricțiilor, (e)... --).*
10. Se consideră problema clasică echilibrată de transport. Există N depozite a unui bun și M consumatori al acelui bun. Sunt cunoscute costurile unitare de transport C_{ij} de transport a unei unitati de bun de la depozitul D_i la consumatorul S_j . Folosiți un algoritm euristic pentru soluționarea acestei probleme. // //
11. Se consideră o problemă de programare matematică aproape liniară, în care restricțiile sunt egalități și inegalități liniare, iar funcția obiectiv este o funcție neliniară. Folosiți un algoritm euristic pentru soluționarea acestei probleme. *(Putem folosi una din metodele de penalizare de rezolvare – (a)nivele fixe de penalizare pt fiecare restricție, (b)nivel dinamic de penalizare, (c)cea mai bună soluție nefeasabilă nu poate fi evaluată mai bine ca cea mai proastă soluție feasibilă, (d)examinarea succesivă a restricțiilor, (e)... --).*
12. Se consideră o problemă de optimizare cu restricții neliniare, iar funcția obiectiv este o funcție liniară. Folosiți un algoritm euristic pentru soluționarea acestei probleme. *(Putem folosi una din metodele de penalizare de rezolvare – (a)nivele fixe de penalizare pt fiecare restricție, (b)nivel dinamic de penalizare, (c)cea mai bună soluție nefeasabilă nu poate fi evaluată mai bine ca cea mai proastă soluție feasibilă, (d)examinarea succesivă a restricțiilor, (e)... --).*

13. Se consideră o problemă concretă de optimizare a unei funcții neliniare în prezența unor restricții arbitrare. Folosiți un algoritm euristic pentru soluționarea acestei probleme. *(Putem folosi una din metodele de penalizare de rezolvare – (a)nivele fixe de penalizare pt fiecare restricție, (b)nivel dinamic de penalizare, (c)cea mai bună soluție nefeasabilă nu poate fi evaluată mai bine ca cea mai proastă soluție feasabilă, (d)examinarea succesivă a restricțiilor, (e)... --).*
14. Să se determine valoarea minimă și punctul de minim al unei funcții polinom de gradul 8 pe segmentul $[a; b]$.
15. Se consideră că o firmă produce 4 tipuri de bunuri, A, B, C, D. La producția lor se folosesc 3 resurse: capital, forță de muncă, timp. Costul fiecărei resurse este de 4, 3, 2, și sunt disponibile în cantitatea 1000, 1200, 900. Pentru producerea unei unități de bunuri de tipul A, B, C, sau D se folosesc corespunzător resursele în următoarele cantități: (3,4,5) pentru A, (2,5,3) pentru B, (5,2,1) pentru C, și (1,3,5) pentru D. Prețul de realizare a unei unități de bun produs A, B, C, sau D este 40, 30, 50, 20 respectiv. Din cauza producției concomitente a bunurilor costurile de producție cresc cu o valoare egală cu $\ln(A) + \ln(B) + \ln(C) + \ln(D)$. Să se determine planul optimal de producție pentru a avea un profit maxim. Folosiți un algoritm euristic pentru soluționarea acestei probleme. *(Putem folosi una din metodele de penalizare de rezolvare – (a)nivele fixe de penalizare pt fiecare restricție, (b)nivel dinamic de penalizare, (c)cea mai bună soluție nefeasabilă nu poate fi evaluată mai bine ca cea mai proastă soluție feasabilă, (d)examinarea succesivă a restricțiilor, (e)... --).*
16. Se consideră o funcție $f(x)$, natura căreia este arbitrară, este definită pe segmentul $(a;b)$ și dorim să o optimizăm (max). // //
17. Se consideră un labirint. Se dorește de a determina ieșirea din labirint folosind algoritmi de inteligență artificială. // //
18. Problema alocării de resurse în spațiu și în timp în anume condiții: problema orarului de muncă - avem consultanți specializați pe anumite probleme, avem clienți care au de rezolvat probleme, fiecare consultant nu poate lucra cu mai mult de 5 clienți la o anumită categorie de probleme. Se dorește de optimizat timpul necesar pentru a asigura consultarea tuturor clienților. Evident, că atât clienții, cât și consultanții au și ei restricțiile lor ce țin de întâlniri. *(Putem folosi una din metodele de penalizare de rezolvare – (a)nivele fixe de penalizare pt fiecare restricție, (b)nivel dinamic de penalizare, (c)cea mai bună soluție nefeasabilă nu poate fi evaluată mai bine ca cea mai proastă soluție feasabilă, (d)examinarea succesivă a restricțiilor, (e)... --).*
19. Etc...

[Studentilor](#) > [Inteligență artificială aplicată](#) > Teme de proiect

(c) andreirusu.eu
