

FIȘA DISCIPLINEI

Metode avansate de rezolvare a problemelor de matematică și informatică

Anul universitar 2026-2027

1. Date despre program

1.1. Instituția de învățământ superior	Universitatea Babeș Bolyai
1.2. Facultatea	Facultatea de Matematică și Informatică
1.3. Departamentul	Informatică
1.4. Domeniul de studii	Informatică
1.5. Ciclu de studii	Licență
1.6. Programul de studii / Calificarea	Inteligență Artificială (Engleză)
1.7. Forma de învățământ	

2. Date despre disciplină

2.1. Denumirea disciplinei		Metode avansate de rezolvare a problemelor de matematică și informatică				Codul disciplinei	MLE5199
2.2. Titularul activităților de curs			Lect. Dr. Mircea Ioan-Gabriel				
2.3. Titularul activităților de seminar			Lect. Dr. Mircea Ioan-Gabriel				
2.4. Anul de studiu	1	2.5. Semestrul	1	2.6. Tipul de evaluare	E	2.7. Regimul disciplinei	obligatorie

3. Timpul total estimat (ore pe semestru al activităților didactice)

3.1. Număr de ore pe săptămână	5	din care: 3.2. curs	2	3.3. seminar/ laborator/proiect	3
3.4. Total ore din planul de învățământ		din care: 3.5. curs		3.6 seminar/laborator/proiect	
Distribuția fondului de timp pentru studiul individual (SI) și activități de autoinstruire (AI)					ore
Studiul după manual, suport de curs, bibliografie și notițe (AI)					
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					
Pregătire seminare/ laboratoare/ proiecte, teme, referate, portofolii și eseuri (mai mare sau egal cu nr. total ore prevăzut în calendarul disciplinei pentru temele de control)					
Tutoriat (consiliere profesională)					
Examinări					
Alte activități [de ex.: comunicare bidirecțională cu titularul de disciplină / tutorele]					
3.7. Total ore studiu individual (SI) și activități de autoinstruire (AI)					
3.8. Total ore pe semestru					
3.9. Numărul de credite					

4. Precondiții (acolo unde este cazul)

4.1. de curriculum	*
4.2. de competențe	*

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	*
5.2. de desfășurare a seminarului/ laboratorului	*

6.1. Competențele specifice acumulate¹

¹ Se poate opta pentru competențe sau pentru rezultatele învățării, respectiv pentru ambele. În cazul în care se alege o singură variantă, se va șterge tabelul aferent celeilalte opțiuni, iar opțiunea păstrată va fi numerotată cu 6.

Competențe profesionale/esențiale	● creează modele de date ● creează softuri ● furnizează documentație tehnică
Competențe transversale	● dă dovadă de inițiativă ● își asumă responsabilitatea ● gândește analitic

6.2. Rezultatele învățării

Cunoștințe	Studentul : ● cunoaște metodele tradiționale de rezolvare a problemelor; ● cunoaște abordările euristice utilizate în situațiile în care metodele tradiționale nu oferă soluții adecvate; ● cunoaște tehnicile de Inteligență Artificială (AI), Învățare Automată (ML) și Învățare Profundă (DL) aplicabile în rezolvarea problemelor NP-complete.
Aptitudini	Studentul: este capabil să implementeze soluții corecte într-un limbaj de programare la alegere; poate optimiza procesul de rezolvare a problemelor, urmărind obținerea soluției optime sau, în cazul problemelor NP-complete, a unei soluții candidate acceptabile; utilizează concepte și tehnici algoritmice pentru a transforma probleme teoretice în soluții funcționale.
Responsabilități și autonomie	Studentul: ● demonstrează capacitatea de a lucra independent, analizând și înțelegând datele de intrare aferente unei probleme; ● este capabil să conceapă algoritmi adecvați pentru rezolvarea problemelor, prin adoptarea, adaptarea și îmbunătățirea tehnicilor predate în cadrul cursului; ● este capabil să evalueze corect rezultatele obținute în urma aplicării algoritmului și să formuleze concluzii privind eficiența soluției propuse.

7. Obiectivele disciplinei (reieșind din grila competențelor acumulate)

7.1 Obiectivul general al disciplinei	● Introducerea studenților în domeniul metodelor de rezolvare a problemelor bazate pe abordări tradiționale, euristice și inteligență artificială (AI).
--	---

7.2 Obiectivele specifice	● Predarea unor tehnici adecvate de rezolvare a problemelor, adaptate la cerințele actuale ale domeniilor informaticii și matematicii. ● Formarea capacității studenților de a alege și aplica metoda optimă (tradițională, euristică sau bazată pe AI) în funcție de tipul și complexitatea problemei abordate. ● Dezvoltarea competențelor de gândire critică și algoritmică necesare pentru proiectarea soluțiilor eficiente.
----------------------------------	--

8. Conținuturi

8.1 Curs	Metode de predare	Observații
Cursul 1. Începutul ● Informație și energie ● Entropie ● Continuum vs Discret ● Static vs Dinamic ● Biți. Logică. Aritmetică. Algebră	Expunere: descriere, explicație, exemple, discuție de studii de caz	
Cursul 2. Algebră și Geometrie ● Geometrie euclidiană vs neeuclidiană ● Design. Principiile designului ● Numere și sisteme numerice	Expunere: descriere, explicație, exemple, discuție de studii de caz	
Cursul 3. Algoritmi și structuri de date tradiționale ● Lumină. Dualitatea undă-particulă ● Căutare ● Structuri de date și algoritmii lor: ○ liniare ○ neciclice ○ ciclice ● Algoritmi și programare ● Hardware și software: Calculatorul	Expunere: descriere, explicație, exemple, discuție de studii de caz	
Cursul 4. Metode de căutare exhaustive vs neexhaustive ● „Should I Stay or Should I Go?” (Ar trebui să rămân sau să plec?) ● Metode algoritmice tradiționale: ○ forță brută / backtracking ○ divide et impera ○ programare dinamică ○ metode greedy (lacome) ● Euristici	Expunere: descriere, explicație, exemple, discuție de studii de caz	
Cursul 5. Algoritmi remarcabili ● Dijkstra. A*	Expunere: descriere, explicație, exemple, discuție de studii de caz	

● Ford-Fulkerson ● Convoluție ● Transformata Fourier Rapidă		
Cursul 5. Fundamentele inteligenței ● Arbori de decizie. Câștig informațional din entropie. Sisteme expert ● Probleme NP-complete. Explorare vs exploatare ● Urcare pe deal (Hill Climbing). Căutare Tabu. Răcire simulată (Simulated Annealing) ● Algoritmi genetici	Expunere: descriere, explicație, exemple, discuție de studii de caz	
Cursul 6. Inteligența colectivă (de tip „stup”) ● Comunități: Optimizarea coloniilor de furnici, Optimizarea roiurilor de particule ● Grafuri și rețele: Problema comis-voiajorului	Expunere: descriere, explicație, exemple, discuție de studii de caz	
Cursul 7. Cum învață mașina ● Teorema limitelor centrale. Statistică ● Metode Monte Carlo ● Gradient descendent. Eroare. Pierdere	Expunere: descriere, explicație, exemple, discuție de studii de caz	
Cursul 8. Înțelegerea cogniției ● Teoria sistemelor ● Modelarea mediului ● Neuronul ● Pradă / Prădător. Sisteme dinamice	Expunere: descriere, explicație, exemple, discuție de studii de caz	
Cursul 9. Învățarea automată I ● Perceptronul ● Învățarea: ○ preprocesarea și vizualizarea datelor ○ antrenarea ○ evaluarea performanței ● Învățarea automată a operațiilor logice și aritmetice de bază	Expunere: descriere, explicație, exemple, discuție de studii de caz	
Cursul 10. Învățarea automată II ● Rețele neuronale ● Propagarea inversă a erorilor (Backpropagation)	Expunere: descriere, explicație, exemple, discuție de studii de caz	

Cursul 11. Învățarea profundă (Deep Learning) ● Vedere: Rețele neuronale convoluționale (CNN) ● Auz: Rețele neuronale recurente (RNN), Memorie pe termen scurt lung (LSTM) ● Învățare prin întărire (Reinforcement Learning) ● Modele Transformer	Expunere: descriere, explicație, exemple, discuție de studii de caz	
Cursul 12. Homo Ludens ● Teoria jocurilor ● Agenți ● Strategii de câștig	Expunere: descriere, explicație, exemple, discuție de studii de caz	
Cursul 13. Logici cuantice ● Logici fuzzy (difuze) ● Dictonomia revizitată: Qubit-ul (bitul cuantic) ● Porți logice cuantice. Calculatorul cuantic	Expunere: descriere, explicație, exemple, discuție de studii de caz	
Cursul 14. Concluzii filosofice și perspective etice		
Bibliografie		
8.2 Seminar / laborator	Metode de predare	Observații
1. Modelarea entropiei 2. Simularea particulelor 3. Simularea geometriei euclidiene 4. Implementarea unei unități aritmetico-logice (ALU) din porți logice 5. Implementarea algoritmilor de căutare 6. Implementarea algoritmilor Dijkstra și A* 7. Implementarea transformatei Fourier rapide (FFT) 8. Implementarea unui algoritm genetic 9. Implementarea unui perceptron 10. Implementarea unei rețele neuronale 11. Implementarea straturilor unei rețele neuronale convoluționale (CNN) 12. Implementarea unui agent 13. Discuție despre computerele cuantice 14. Implementarea unui joc complet		
Bibliografie		
1. DONALD E. KNUTH, The Art of Computer Programming, Addison-Wesley, 1998 2. P. N. Klein, Coding the Matrix. Linear Algebra through Applications to Computer Science, Newtonian Press, 2013. 3. P.D. Lax, M.S. Terrell, Calculus with Applications, Springer, 2014. 4. S. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, Prentice Hall, 1995 5. KERNINGHAN B.W., RITCHIE D.M The C Programming Language -- Ansi C. Prentice Hall, 1988.		

9. Coroborarea conținuturilor disciplinei cu așteptările reprezentanților comunității epistemice, asociațiilor profesionale și angajatori reprezentativi din domeniul aferent programului

- Absolventul cunoaște, înțelege și aplică conceptele de bază și algoritmi fundamentali ai Inteligenței Artificiale și este capabil să îi evalueze pe baza unor metrici.
- Absolventul cunoaște și înțelege conceptele și tehnicile de reprezentare a cunoștințelor și este capabil să le aplice în rezolvarea problemelor.
- Absolventul cunoaște și înțelege fundamentele matematice necesare dezvoltării algoritmilor inteligenți și este capabil să le utilizeze în implementarea acestora.
- Absolventul este capabil să evalueze, atât cantitativ cât și calitativ, performanța sistemelor inteligente.
- Absolventul este capabil să aplice algoritmi fundamentali ai Inteligenței Artificiale pentru a rezolva probleme din lumea reală.
- Absolventul cunoaște și înțelege fundamentele matematice necesare dezvoltării algoritmilor inteligenți și este capabil să le utilizeze în implementarea acestora. *(Notă: fraza apare de două ori în textul original — o pot păstra o singură dată dacă vrei.)*
- Absolventul este capabil să combine informații diverse pentru a formula soluții și a dezvolta idei pentru noi produse și aplicații.

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs		examinare orală	60%
10.5 Seminar/laborator		activități în cadrul seminariilor/laboratoarelor	40%
10.6 Standard minim de performanță			
● Fiecare student trebuie să fie capabil să utilizeze tehnici tradiționale de rezolvare a problemelor și metode euristice pentru soluționarea problemelor de matematică și informatică. ● Pentru promovarea disciplinei, studentul trebuie să obțină o notă finală de minimum 5 (cinci), calculată prin cumularea punctajelor obținute din activitățile de laborator / seminar, și examinarea orală finală din sesiunea de examene.			

11. Etichete ODD (Obiective de Dezvoltare Durabilă / Sustainable Development Goals)²

Nu se aplică.

Data completării:

Semnătura titularului de curs

Semnătura titularului de seminar

...

Lect.dr. Mircea Gabriel

Lect.dr. Mircea Gabriel

² Păstrați doar etichetele care, în conformitate cu [Procedura de aplicare a etichetelor ODD în procesul academic](#), se potrivesc disciplinei și ștergeți-le pe celelalte, inclusiv eticheta generală pentru *Dezvoltare durabilă* - dacă nu se aplică. Dacă nicio etichetă nu descrie disciplina, ștergeți-le pe toate și scrieți "Nu se aplică".

Data avizării în departament:

...

Semnătura directorului de departament

Conf.dr. Adrian STERCA