***Abstract***

Problema de timetabling în cadrul universităților este o problemă NP-hard, care presupune programare unor sarcini pentru anumite persoane avînd la bază multiple constrângeri și o serie de resurse limitate. Această lucrare abordează problema programării orelor academice (sau a orarului). Scopul este de a găsi un orar fezabil pentru facultatea de Economie, Administraţie şi Afaceri din Universitatea „Ştefan cel Mare” din Suceava. Abordarea se concentrează pe utilizarea algoritmilor genetici şi are ca principal obiectiv minimizarea pauzelor studenţilor participanţi la activităţilor didactice

**Introducere**

UCTTP(University Course TimeTabling Problem) este o problemă de optimizare în clasa de probleme NP-hard. Această activitate apare la începutul fiecărui semestru în cadrul universităților și include alocarea de evenimente (cursuri, profesori și studenți) unui număr de intervale de timp și săli fixe. Această problemă trebuie să satisfacă atât constrângerile hard, cât și constrângerile soft în timpul alocării evenimentelor către resurse. În ceea ce priveşte constrângerile hard, ele trebuie să fie îndeplinite în totalitate, iar pentru creşterea calităţii orarelor se doreşte o îndeplinire a cât mai multor contrângeri soft enumerate în cadrul problemei[[1]](#footnote-1) [[2]](#footnote-2).

Constrângerile hard ce ar putea să apară într-o problemă de optimizare din categoria tratată în această lucrare, ar putea fi următoarele[[3]](#footnote-3):

* Nu se poate depăşi intervalul orar 8-20;
* Nu se pot pune ore în alte zile decât cele 5 destinate activităţilor didactice(de luni până vineri);
* Un profesor nu poate ţine două cursuri diferite în acelaşi interval de timp;
* Un student nu poate participa la două activităţi diferite în acelaşi interval de timp;
* Numărul de studenţi prezenţi la o activitate nu trebuie să depăşească numărul de locuri din sală;
* Activităţile pot fi plasate doar în săli destinate acelor categorii de activităţi pentru care au fost destinate;
* Numărul maxim de ore de curs pentru un student pe zi va fi de 6, în timp ce numărul maxim total de activităţi didactice va fi de 8;

Constrângerile soft ce ar putea fi întâlnite în cadrul unei astfel de probleme sunt: profesorul ar putea opta pentru nişte intervale în care să-şi desfăşoare activitatea, profesorul ar putea specifica sala în care să-şi desfăşoare activitatea, se poate acorda pauză de masă (la prânz) persoanelor care încep orele de la 8 dimineaţă, etc.

Printre abordările utilizate, până în prezent, pentru rezolvarea problemei UCTTP enumerăm următoarele: tehnici din cercetări operaţionale (colorarea grafurilor[[4]](#footnote-4), programarea liniară[[5]](#footnote-5)), algoritmi genetici[[6]](#footnote-6) [[7]](#footnote-7), ant colony optimization[[8]](#footnote-8), memetic algorithm[[9]](#footnote-9), harmony search algorithm[[10]](#footnote-10), particle swarm optimization[[11]](#footnote-11), tabu search[[12]](#footnote-12), dar şi variante de optimizare multicriterială[[13]](#footnote-13) [[14]](#footnote-14).

În cadrul acestei lucrări am recurs la descrierea rezultatelor obţinute în cadrul UCTTP, problemă rezolvată folosind algoritmii genetici. Rezultatele descrise au fost obţinute prin rularea unui program scris în Java, folosind tehnologia orientată obiect, cu date din cadrul facultăţii de Economie, Administraţie şi Afaceri din Universitatea „Ştefan cel Mare”Suceava.

**Descrierea problemei**

Problemele de timetabling educaţional pot fi de mai multe tipuri: programarea examenelor universitare, programarea cursurilor universitare, programarea în cadrul diferitelor proiecte şi altele. Dificultăţile care apar în dezvoltarea unor astfel de programări sunt din ce în ce mai numeroase. Instituţiile de învăţământ superior introduc din ce în ce mai multe noi specializări, creeză noi facultăţi, ceea ce duce la creşterea numărului de studenţi. De cele mai multe ori, viteza de creştere a acestor elemente este net superioară creşterii resurselor materiale (în principal, spaţii de învăţământ). Acest lucru duce la creşterea numărului de restricţii ce se impun a fi respectate în momentul în care se încearcă crearea unui orar universitar. De exemplu, în cadrul Universităţii „Ştefan cel Mare” din Suceava, există 11 facultăţi, cu număr diferit de specializări la programele de licenţă. Astfel, de la facultăţi care au două specializări, cu un total de 12 grupe, pentru 4 ani de studiu, ajungem la Facultatea de Economie, Administraţie şi Afaceri la un total de 53 de grupe, pentru 3 ani de studiu. Această creştere a numărului de grupe duce la un efort considerabil, în momentul în care se doreşte crearea orarului pentru studenţi. Fiind o problemă de optimizarea combinatorială destul de dificilă, rezolvarea acesteia folosind prelucrări manuale este destul de dificilă, resursa timp necesară fiind risipită, indiferentă de mărimea resursei umane implicate în această activitate de programare. Acest considerent ne obligă să recurgem la găsirea de soluţii informatice în rezolvarea problemei.

Dintre metodele ce se utilizează pentru rezolvarea problemei de optimizare descrise, am recurs la tehnici bazate pe populaţii, şi anume algoritmi genetici. Algoritmii genetici sunt utilizaţi în diverse domenii pentru rezolvarea problemelor de optimizare. Ei sunt inspiraţi din procesul de evoluţie naturală, acest lucru fiind întărit şi de denumirea pe care o au. Algoritmii lucrează cu o populaţie de posibile soluţii şi includ trei paşi esenţiali[[15]](#footnote-15): selecţia, regenerarea şi înlocuirea. În faza de selecție, indivizii cu fitness ridicat sunt selectaţi pentru a fi părinți pentru următoarea generație. În faza de regenerare se utilizează doi operatori: de încrucișare și de mutație asupra părinților care au fost selectați în prima fază și în faza de înlocuire: indivizii din populația inițială sunt înlocuiți cu indivizii nou creați.

Algoritmii genetici, care sunt un subset de EA (Evolutionary Algorithms), se bazează pe următorii pași: (1) generarea populației inițială, (2) evaluarea populației generate folosind funcția de evaluare, (3) selectarea părinților pentru a se încrucișa având la bază informațiile obținute din funcțiile de evaluare, (4) se aplică operatorul de încrucișare pentru a produce copii, (5) se aplică operator de mutație pentru a produce copii, (6) se selectează părinții și copiii pentru a forma noua populație pentru generația viitoare și (7) dacă condiția de terminare este îndeplinită, algoritmul se oprește, altfel trece la pasul doi(generaţia următoare) și continuă[[16]](#footnote-16).

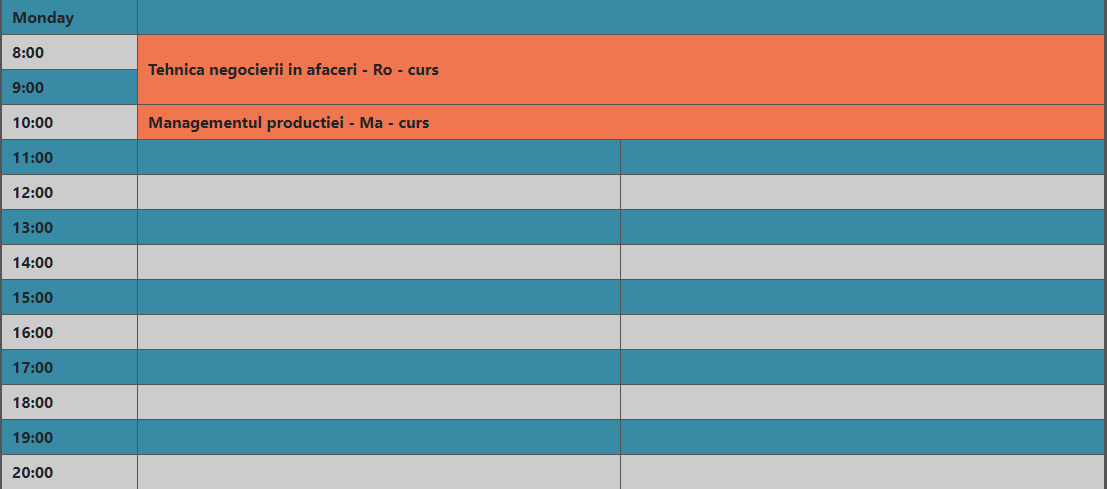
Populaţia utilizată în cadrul testelor a căror rezultat este prezentat în cadrul acestui articol este formată din 100 cromozomi, ce vor evolua timp de 200 de generaţii. În fiecare generaţie vom aplica 2 operatori diferiţi de mutaţie(primul de 20 de ori, iar al doilea de 10 ori) şi unul de încrucişare (de 20 de ori).

Un cromozom are dimensiunea egală cu numărul de activităţi ce vor fi introduse în orar (vom avea 523 de activităţi didactice ce vor trebui introduse în orar), la care se adaugă un element ce va memora valoarea funcţiei obiectiv. O genă ne va furniza o serie de informaţii, precum: denumirea specializare, disciplina, titularul, anul de studiu, grupa, numărul de studenţi, dacă e curs sau nu, număr de săptămâni în care se ţine activitatea, număr de ore pe săptămână, număr de grupe şi semigrupe de la acea specializare.

**Computational results**

Problema ce se doreşte a fi optimizată are ca date de intrare cursurile şi activităţile pratice(seminarii şi laboratoare) din cadrul Facultăţii de Economie, Administraţie şi Afaceri din cadrul Universităţii „Ştefan cel Mare” Suceava. Datele luate în considerare sunt aferente semestrului II din anul universitar 2019-2020. Numărul de specializări pentru care se va efectua optimizarea este 9, iar pentru fiecare specializare avem 3 ani de studiu. În cadrul instituţiei activează 63 de cadre didactice, care pot avea de la o singură activitate pe săptămână, până la peste 25 de activităţi pe săptămână. Activităţile aferente problemei studiate se desfăşoară de luni până vineri, de la ora 8:00 la 20:00. Activităţile de curs se ţin cu un an aferent unei anumite specializări, cele se seminarii cu o grupă ce aparţine unei specializări, iar laboratorarele se ţin la nivel de semigrupă. Din afirmaţia precedentă rezultă că în momentul în care la o anumită specializare se desfăşoară activităţi de curs, nu se pot desfăşura activităţii de seminarii sau laboratoare.

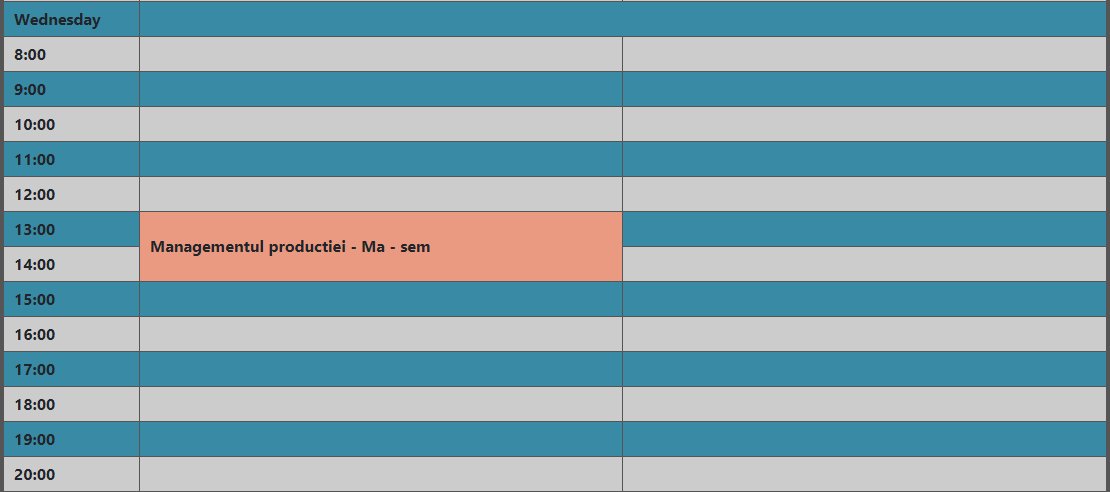
În cazul de faţă doresc să optimizez the idle time aferent resursei „studenţi”. Această resursă umană are o importanţă mare pentru instituţia pentru care se generează orarul, de aceea în cadrul acestei abordări am pus în centrul atenţiei această categorie. Funcţia obiectiv depinde de numărul de ore libere avute de către studenţi între două activităţi ce se desfăşoară în aceeaşi zi. Studenţii pot fi împărţiţi la activităţi în 3 mari categorii de grupuri: anul întreg pentru cursuri, grupa pentru seminarii şi semigrupa pentru laboratoare. Calcularea valorii idle time pentru fiecare specializare s-a făcut prin însumarea orelor libere, aflate între activităţile didactice, la nivel de semigrupă.



a.



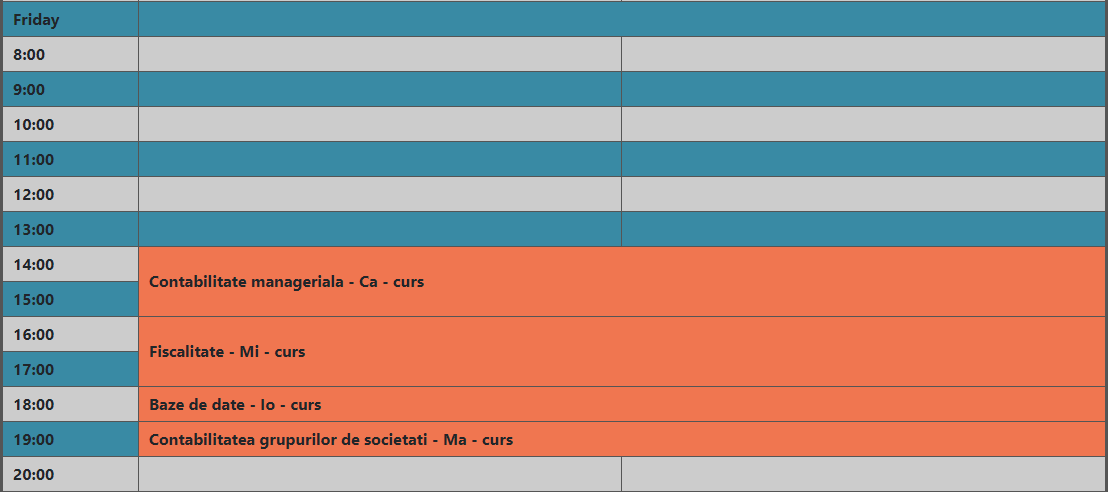
b.



c.



d.



e.

Figura 1 Orarul săptămânal pentru o specializare

În figura 1 prezint orarul la o specializare(notată AF3, ce are în componenţă 2 grupe şi 2 semigrupe). Se observă că idle time este mic, fiind dat de cele 3 ore, din ziua de marţi, la prima grupă, între ora 12:00 şi 15:00. Desigur, specializarea aleasă pentru a fi reprezentată nu este chiar aşa de complexă. La alte specializări ce au în componenţă mai multe grupe(şi semigrupe) numărul de ore de pauză va creşte. În lucrarea „A new genetic approach for Course Timetabling Problem”[[17]](#footnote-17), am prezentat atât componenţa fiecărei specializări, cât şi evoluţia funcţiei ce se doreşte a fi optimizată (în cele 200 de generaţii), dar si orarul unei specializări formate din 2 grupe şi 3 semigrupe (unde se observă mai multe ore de pauză). Deşi, la formaţiile de studiu mai numeroase apar mai multe pauze nedorite, cu ajutorul algoritmului dezvoltat am ajuns la performanţa, să obţinem după cele 200 de generaţii, un număr total de ore ce reprezintă idle time la valori de sub 200.

Prin rezultatele obţinute am atins obiectivul principal propus, acela de a minimiza idle time pentru studenţi. Totuşi, pe lângă această resursă umană, în cadrul instituţiei, avem si resurse profesori, pentru care nu s-a încercat o optimizare a acesui idle time. Repartizarea orelor către cadrele didactice titulare nu este uniformă. Astfel, anumiţi profesori au de susţinut mai multe activităţi, în timp ce alţii susţin două, sau poate numai o singură activitate săptămânală. Ţinând cont că, în cadrul acestei lucrări, accentul s-a pus pe studenţi, repartizarea cadrelor didactice în orar poate crea nemulţumiri pentru această din urmă categorie. În figura 2 am prezentat legătura dintre numărul de ore pe care îl are fiecare cadru didactic şi numărul de zile dintr-o săptămână, în care aceştia îşi susţin activităţile.

Figura 2. Corelaţia dintre numărul de ore săptămânale şi numărul de zile din săptămână necesare susţinerii activităţii

Din această figură putem observa, că în marea majoritatea a cazurilor repartizarea orelor pentru fiecare cadru didactic se face echitabil. Mici excepţii sunt sesizate în anumite cazuri (excepţii evidenţiate în dreptunghiuri). Astfel, s-a ajuns ca un cadru didactic care are de efectuat 6 ore săptămânal să aibă nevoie doar de o singură zi a săptămânii pentru a-şi rezolva sarcinile, în timp ce altă persoană care trebuie să susţine 4 ore de activităţi didactice să fie obligată să vină în 3 zile la locul de muncă (exemple evidenţiate, în cadrul figurii, prin cerc).

Figura 3. Corelaţia dintre numărul de ore săptămânale şi pauzele aferente fiecărui cadru didactic

În figura 3 avem reprezentată legătura dintre numărul de ore săptămânale aferente fiecărui cadru didactic şi idle time pentru fiecare în parte. Se observă şi în acest caz mici problem. Astfel, se ajunge ca un professor care trebuie să susţină 30 de ore de activităţi didactice să aibă numai 3 ore de pauză, altul care are 24 de ore de activitate să aibă 25 de ore de pauză, iar al treilea la cele 9 ore de activitate să aibă 10 ore de pauză. Toate cele trei cazuri sunt evidenţiate, în cadrul figurii, în dreptunghiuri.

Pentru îmbunătăţirea acestor ultime rezultate, fără a afecta rezultatele obţinute pentru student, se recomandă o abordare multi-criterială pentru problema studiată. În cadrul acestei abordări se pot folosi, de asemenea algoritmii genetici. Optimizarea idle time aferent profesorilor poate fi făcută în acelaşi timp cu cea aferentă studenţilor, un lucru nou ce ar putea să apară, putând fi considerat ca o constrângere soft, ar fi preluarea opţiunilor de orar de la cadrele didactice.

**Concluzii**

În cadrul acestei lucrări prezentăm rezultatele obţinute folosind algoritmi genetici, pentru o problema de timetabling universitar. Studiul de caz a fost o facultate din cadrul Universităţii “Ştefan cel Mare” Suceava. Accentul s-a pus pe resursa umană constituită din studenţi. Ţinând cont de complexitatea unei astfel de problem, rezultatele obţinute au fost destul de bune, timpii de rulare au fost satisfăcători iar resursele computaţionale utilizate nu au fost cu un grad ridicat de performanţă (Procesor Intel core i7, 12 GB RAM). Această abordare computaţională aduce un beneficiu real organizaţiei, faţă de variant clasică, cea manuală, care presupunea alocarea pentru această activitate a unui anumit număr de personae, care ar trebui, în funcţie de complexitate, să aloce multe ore pentru îndeplinirea sarcinilor. Rezultatele obţinute folosind algoritmii descrişi sunt net superiori variantei clasice. Totuşi, studentul nu este singura resursa umană implicată în activităţile didactice. Chiar dacă se doreşte un învăţământ centrat pe student, trebuie să avem în vedere şi resursa umană didactică (profesorii). O propunere, pentru satisfacerea ambelor categorii de persoane implicate în procesul educational, ar fi utilizarea optimizării multiobiectiv, abordare ce va duce la o echilibrare a idle time atât pentru studenţi, cât şi pentru profesori. Această nouă abordare va fi dezbătută ţinând cont şi de preferinţele de orar atrebuite profesorilor.

1. Asmuni, H., Fuzzy methodologies for automated university timetabling solution construction and evaluation, Ph.D. Thesis, School of Computer Science University of Nottingham, 2008 [↑](#footnote-ref-1)
2. Obit, J. H., Developing novel meta-heuristic, hyper-heuristic and cooperative search for course timetabling problems. Ph.D. Thesis, School of Computer Science University of Nottingham, 2010 [↑](#footnote-ref-2)
3. Balan, I., A new genetic approach for Course Timetabling Problem, Journal of Applied Computer Science &Mathematics, nr.1, Vol.15, 2021 [↑](#footnote-ref-3)
4. Trio Budiono, Kok Wai Wong, A pure graph coloring constructive heuristic in timetabling, Conference: International Conference on Computer & Information Science (ICCIS), 2012 [↑](#footnote-ref-4)
5. Ranga Prabodanie, An Integer Programming Model for a Complex University Timetabling Problem: A Case Study, Industrial Engineering & Management Systems 16(1):141-153, 2017 [↑](#footnote-ref-5)
6. Prof-Othman Alsmadi, Zaer Abo-Hammour, Dia I. Abu-Al-Nadi, Alia Algsoon, A novel genetic algorithm technique for solving university course timetabling problems, 2011 [↑](#footnote-ref-6)
7. Balan, I., A new genetic approach for Course Timetabling Problem, Journal of Applied Computer Science &Mathematics, nr.1, Vol.15, 2021 [↑](#footnote-ref-7)
8. Munirah Mazlan, Mokhairi Makhtar, Ahmad Firdaus Khair Ahmad Khairi, Mohamad A Mohamed, University course timetabling model using ant colony optimization algorithm approach, Indonesian Journal of Electrical Engineering and Computer Science 13(1):72-76, 2019 [↑](#footnote-ref-8)
9. Sadaf N. Jat, Shengxiang Yang, A Memetic Algorithm for the University Course Timetabling Problem, 20th IEEE International Conference on Tools with Artificial Intelligence, 2008 [↑](#footnote-ref-9)
10. Mohammed Azmi Al-Betar, Ahamad Tajudin Khader, A harmony search algorithm for university course timetabling, Annals of Operations Research 194(1):3-31, 2012 [↑](#footnote-ref-10)
11. Ho Sheau Fen Irene, Safaai Deris, Siti Zaiton Mohd Hashim, University course timetable planning using hybrid particle swarm optimization, GEC '09: Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation, 2009 [↑](#footnote-ref-11)
12. Tanzila Islam, Zunayed Shahriar, Mohammad Anower Perves, Monirul Hasan, University Timetable Generator Using Tabu Search, Journal of Computer and Communications > Vol.4 No.16, 2016 [↑](#footnote-ref-12)
13. Vertic Eridani Budi Darmawan, Yuh-Wen Chen, Aisyah Larasati, Anik Dwiastuti, Multi-objective Modeling for a Course Timetabling Problem, International Conference on Creative Economics, Tourism & Information Management, 2019 [↑](#footnote-ref-13)
14. Md Shahriar Mahbub, Shihab Shahriar Ahmed, Kazi Irtiza Ali, Md. Taief Imam, A Multi-Objective Optimization Approach for Solving AUST Classtimetable Problem Considering Hard and Soft Constraints, I. J. Mathematical Sciences and Computing, 5, 1-14, 2020 [↑](#footnote-ref-14)
15. Hamed Babaei, JaberKarimpour, Amin Hadidi, A survey of approaches for university course timetabling problem, Computers & Industrial Engineering, Volume 86, Pages 43-59, 2015 [↑](#footnote-ref-15)
16. Obit, J. H., Developing novel meta-heuristic, hyper-heuristic and cooperative search for course timetabling problems. Ph.D. Thesis, School of Computer Science University of Nottingham, 2010 [↑](#footnote-ref-16)
17. Balan, I., A new genetic approach for Course Timetabling Problem, Journal of Applied Computer Science &Mathematics, nr.1, Vol.15, 2021 [↑](#footnote-ref-17)