

# Tema 1: Orar - IA 2024

*Paiu Teofil 333CB*

## **Cuprins:**

<b>Introducere.....</b>	<b>2</b>
<b>Reprezentare Problema.....</b>	<b>2</b>
<i>Hill Climbing</i> .....	2
Abordare Stochastica.....	2
Reprezentarea stărilor.....	2
Rezultate obținute pe parcurs.....	3
<i>Partial Constraint Satisfaction Problem</i> .....	3
Reprezentare Variabile.....	3
Reprezentare Domeni.....	3
Reprezentare Constrangeri.....	3
Rezultate obținute pe parcurs.....	4
<b>Analiza rezultate finale.....</b>	<b>4</b>
<i>Rezultate Hill Climbing</i> .....	4
<i>Rezultate PCSP</i> .....	4
<b>Concluzie.....</b>	<b>4</b>

---

# Introducere

În acest document o sa prezint abordarea, modelarea problemei și rezultatele obtinute in urma semi realizării primului assignment al cursului de Inteligență Artificială.

Assignment-ul respectiv a avut loc in anul universitar 2023-2024 în al doilea semestru și a fost bazat pe implementarea a 2 algoritmi de căutare (Hill Climbing si PCSP) pentru a rezolva o problema cunoscută ci anume asignarea de valori pentru intervalele unui orar.

---

## Reprezentare Problema

### Hill Climbing

#### Abordare Stochastica

Pentru căutarea locală a soluției am ales sa folosesc o implementarea stochastica modificată a acestui algoritm deoarece are șanse de a reuși mai mari față de cea simpla și de celelalte doua implementări (first-choice sau best-first).

Modificare a rezultat în a micșora numărul de iterații total, aceasta se bazează pe o alegere stocastica cat timp exista doar stări vecine egale ca și cost cu cea curentă se alege aleator una dintre acestea pentru a explora 'platoul', iar în caz ca exista stării mai bune se alege parcurgerea lor fata de explorarea platoului.

#### Reprezentarea stărilor

Starea este reprezentată de o clasa care are 3 atribut principale și anume o reprezentare identică a orarului ca cea a funcțiilor de afișare date, datele fișierului de intrare citite cu funcția "read\_yaml\_file()" și numărul de constrangeri încălcate reprezentate ca un tuplu cu primul element reprezentand constrângerile hard încălcate, iar al doilea constrângerile soft încălcate.

Starea inițială este initializata cu un orar gol, specificatile orarului și constrângerile actuale ale orarului gol. O stare  $Q_j$  vecină a stării curente  $Q_i$ , este o stare care diferă de  $Q_i$  printr-o singura schimbare a unei alocări din orarul stării  $Q_i$ . Unde unei combinatii dintre zi, interval, curs i se alocă profesor, materie sau nimic.

## Rezultate obținute pe parcurs

Inițial fără abordarea modificată a platoul algoritmul avea o sansa de castig de aproximativ 40 % (rezultat luat din rulara succesivă a mai multor teste orar\_mic\_exact), deoarece acesta nu găsea un punct de minim local in limita iteratiilor date.

După implementarea modalității de a ieși din platourile intalnite pe parcurs acuratețea algoritmului a crescut semnificativ (80% rezultat luat din rulara succesivă a mai multor teste orar\_mediu\_exact) ramanand o singura problema ci anume complexitatea temporală.

Pentru a rezolva aceasta problema am ales sa reduc spațiul de căutare al algoritmului din implementarea inițială care îmi genera toate posibilitatile de schimbare printr-o mișcare a stării curente, cu alegerea aleatoare a N zile din orar (N fiind egal 3 pentru zile mai mici decat 3 și 2 în rest, deoarece în cazul problemei dummy aceasta configurație limita prea mult informațiile disponibile la fiecare iterație ducand la pierderea acurateței) astfel reducand nivelul de informație al fiecărui nivel și scazand complexitatea de generare.

---

## Partial Constraint Satisfaction Problem

### Reprezentare Variabile

Pentru reprezentarea variabilelor am ales aceeași abordare ca la hill climbing ci anume combinati de zile, intervale și săli aceasta sa dovedit a fiind ineficienta deoarece numărul total de combinatii generabile cu valorile posibile acestor variabile este mult prea mare din punct de vedere computational.

### Reprezentare Domeni

Domenile sunt reprezentate de combinati de profesor, curs asiginate doar în sălile în care materia respectiva se poate preda.

### Reprezentare Constrangeri

Pentru constrangeri nu am găsit o modalitate buna de a le prioritiza pe cele logice (hard) și problema principala este ca majoritatea dintre aceste constrangeri se verifica la final unde este nevoie de o valoare pentru toate variabilele pentru a calcula de exemplu acoperirea unui curs sau numărul de ore al unui profesor.

Acestea au aceeași reprezentare ca cea prezentată la laborator cu o mica excepție, deoarece pentru calcularea acoperiri studentilor la o materie aveam nevoie sa stiu un atribut ce făcea parte din variabila și funcția de calculare a constrangeri trebuia sa primeasca acel atribut am creat un al 3-lea element in tuplu fiind un flag care imi indica și parsarea variabilelor în funcție pe langa valorile acesteia.

## Rezultate obținute pe parcurs

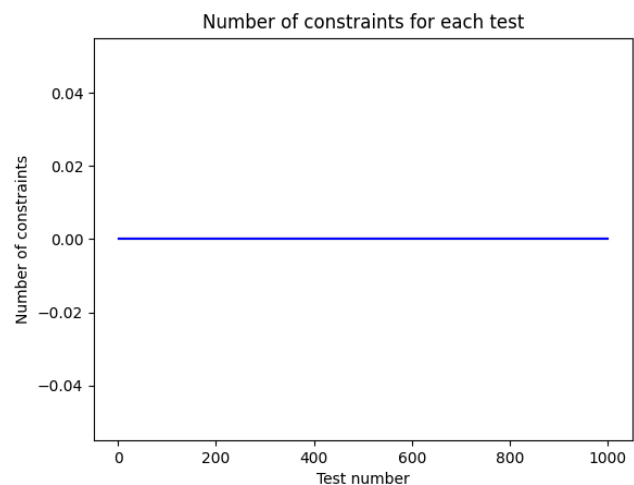
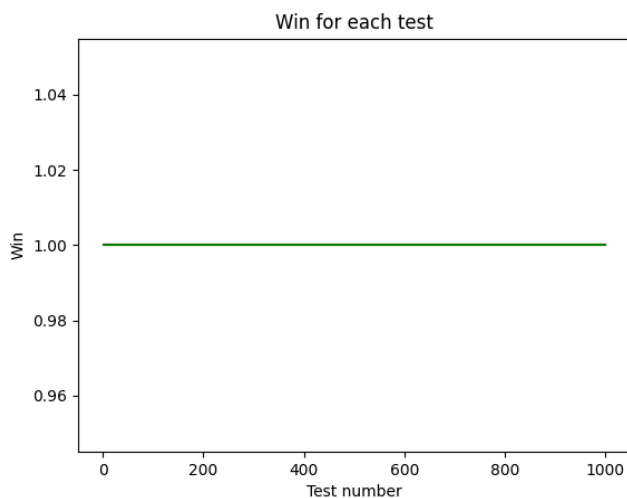
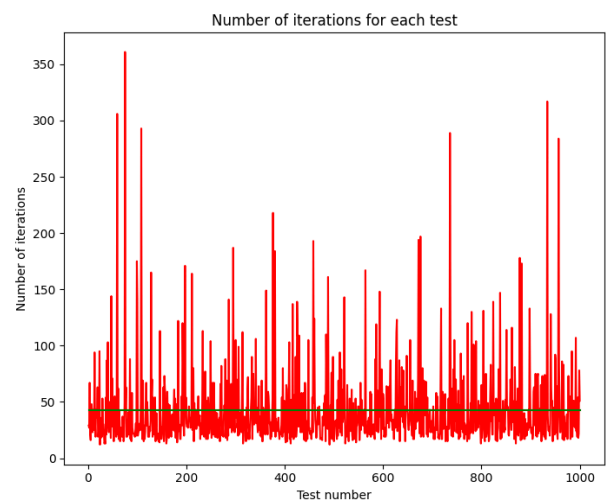
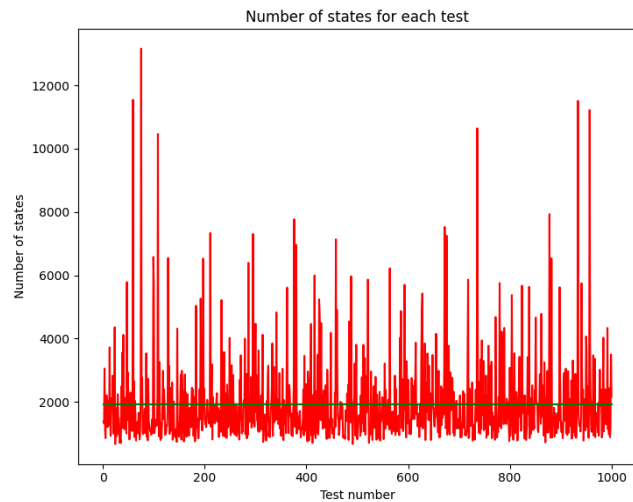
Pentru început am scos partea de cost acceptabil pentru a vedea posibile soluții găsite pe parcurs cele mai bune fiind de cost 18, după introducerea condiției de cost acceptabil 0 algoritmul continuă să caute recursiv soluția dar cu o complexitate temporală extrem de mare .

## Analiza rezultate finale

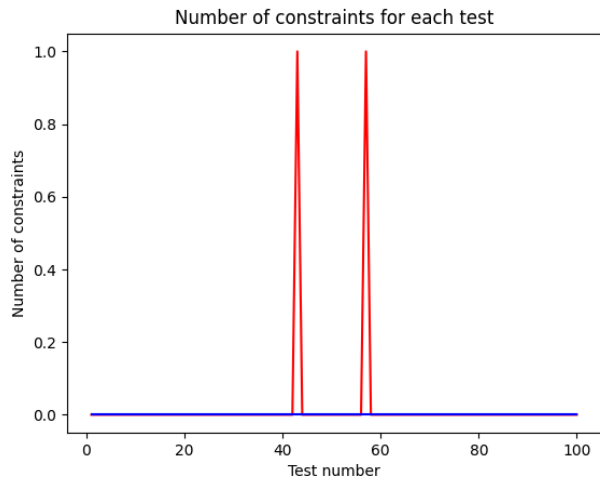
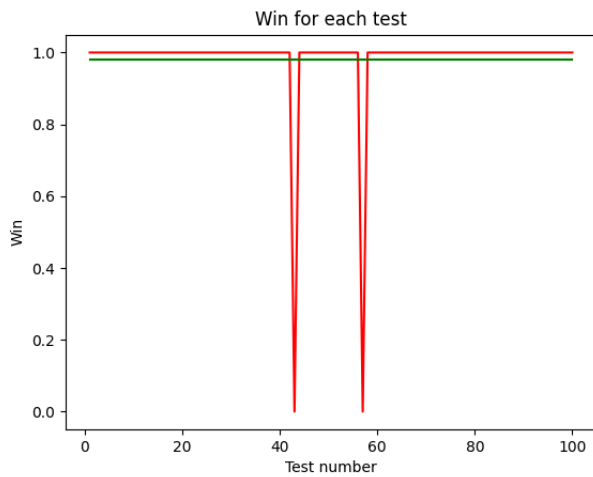
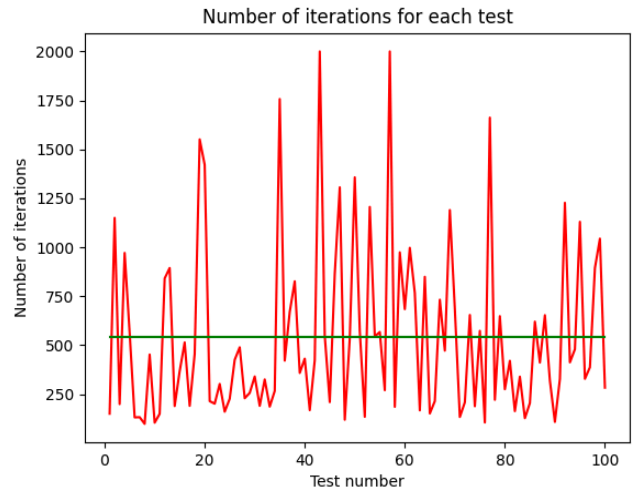
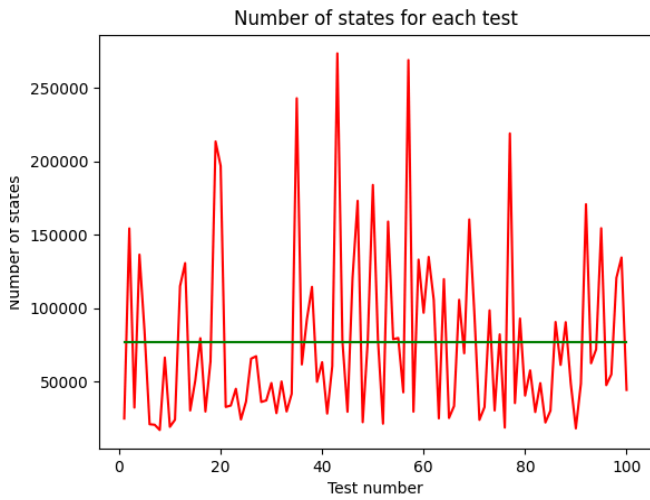
### Rezultate Hill Climbing

În urma unor teste am obținut aceste rezultate pe următoarele teste:

a) Dummy (1000 de rulari)



b) Orar\_mic\_exact (100 rulari)



Pentru primele 3 grafice linia verde reprezinta media valorilor iar pentru ultimul linia rosie este numărul de constrangeri hard încălcate, iar cea albastra de cele soft.

Pentru cazul testele orar\_mic\_exact picate sunt sigur ca cu un numar mai mare de iterati acestea ar fi trecut dar am limitat rulara unui test la aproximativ un minut (acesta ruleaza in medie 22 de secunde).

Pentru testele dummy timpul total a fost 8 minute, iar pentru cele de orar\_mic\_exact 36 de minut

## Rezultate PCSP

Din păcate nu am avut suficient timp în aceasta perioada pentru a găsi o soluție problemei. Iar din aceasta cauza nu am primit rezultate finale. Cred ca o abordare diferită și o încercare de a privi și modela problema în subprobleme care sa poate fi evaluate pentru constrângerile hard la fiecare pas.

## Concluzie

Din motive mi-a fost mai ușor sa modelez cerințele problemei pe o căutare locală și reprezentarea acesteia ca o stare. Cred ca în final am înțeles mult mai bine funcționarea ambilor algoritmi și că trebuie sa mai experimentez cu pcsp pana la examen.