# Tema 1 IA

#### Setup

#### State: o clasa care contine

- O matrice care contine un dictionar cu clasele alocate pentru fiecare interval
- Un dictionar cu toate materiile si cati studenti mai sunt de alocat
- Preferintele profesorilor matrice in care fiecare interval e marcat cu 1 daca este preferat,
   0 daca nu este si –1 daca este alocat deja
- Un dictionar in care pentru fiecare sala este marcat intr-o matrice daca este sau nu folosita in fiecare interval
- Un dictionar cu materiile pe care le preda fiecare profesor (Putea sa fie o instanta globala)
- Un dictionar care contorizeaza cate ore alocate are fiecare profesor (ca sa nu treaca de 7)
- Numarul de conflicte actuale (constrangeri incalcate)

## Optimizari

- Se aloca ore la materii in ordinea crescatoare a numarului de studenti inscrisi la materiile respective
- Random choice, random restart

## Hill Climbing

La generarea state-urilor urmatoare, se genereaza state-uri care incalca constrangeri noi
doar cat timp nu se gaseste un next state care nu incalca nicio constrangere, caz in care se
vor lua in considerare doar state-urile care nu adauga conflicte

#### Monte Carlo Tree Search

- Se incearca generare doar de stari fara conflicte, pana se ajunge la un dead end, caz in care no\_conflicts\_possible trece pe False si se accepta stari care adauga conflicte
- Cand la parcurgere se ajunge intr-o stare cu mai multe conflicte decat cel mai bun state final gasit, se opreste parcurgerea si se trece la o noua iteratie
- Reward-ul pentru o stare blocata este 0
- Reward-ul pentru stari finale este o sigmoida "inversa", este de preferat cazul cu 0 conflicte si sustinut sa ai cat mai putine, rewardul tinde la 0 cu cat cresc conflictele

# Hill Climbing

#### Dummy

```
**adigAdi-OmenNP:-/Uni/IA/Temals python3 orar.py hc dummy

**adigAdi-OmenNP:-/Uni/IA/Temals

**adigAdi-OmenNP:-/Uni/IA/Temals
```

#### Orar Mic

```
# Seal Control | Deliver |
```

#### Orar Mediu

#### Orar Mare

#### **Orar Constrans Incalcat**

```
adiaAdi-OmeniP:-/Uni/IA/Temais python3 orar.py hc orar_constrans_incalcat
Total states: 33335
Restorts: 10
Re
```

## Monte Carlo Tree Search

#### Dummy

```
### BROBLEMS 1 OUTPUT DEBUCCONSOLE TERMINAL PORTS JUPPTER

#### addition_demNPy-/Uni/IA/Temal$ python3 orar.py mcts dummy
Total states: 6
Total states: 12
Total states: 15
Total states: 15
Total states: 16
Total states: 19
Total states: 10
Tota
```

#### Orar Mediu

```
PROBLEMS 1 OUTPUT DEBUGCONSOLE TERMANAL PORTS JUPYTER

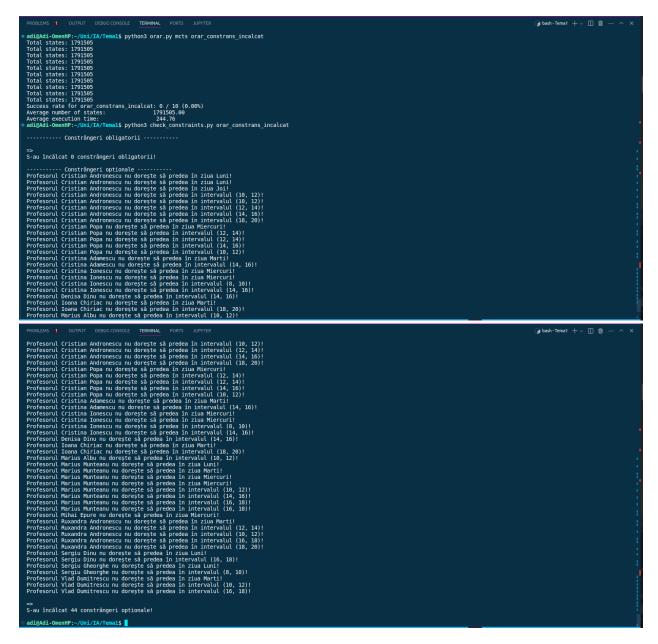
addigAddi-OmenHP:-/Uni/IA/Temal$ python3 orar.py mcts orar_mediu_relaxat

Total states: 356

Total states: 356
```

#### Orar Mare

#### Orar Constrans Incalcat



Pentru scenarii la care se poate ajunge la rezolvare fara conflicte

- MCTS este mult mai rapid decat HC si genereaza de 5-10 ori mai putine stari
- Ambii algoritmi au 100% success rate

Pentru scenarii care nu accepta rezolvare fara conflicte (sau au un grad de complexitate ridicat)

- HC gaseste in medie de intr-un minut solutii cu un numar scazut de conflicte
- Pe orar\_constrans\_incalcat -> in medie 10 conflicte

- MCTS gaseste in rulari de in medie 6 minute solutii cu de 5 ori mai multe conflicte decat HC; cresterea (substantiala) a bugetului nu a parut a aduce imbunatatiri in aceasta directie
- HC genereaza si de 60 de ori mai putine stari decat MCTS in parametrii actuali de rulare