1) Reprezentarea starilor si a restrictiilor

O stare are urmatoarea structura:

- board
- nconflicts

“board” este un timetable de tipul:

board : {str : {(int, int) : {str : (str, str)}}}

(board : {ziua : {(interval1, interval2) : {sala : (profesor, materie)}}})

“nconflicts” este un int ce reprezinta suma costurilor conflictelor din timetable-ul starii

Algoritmul porneste de la un timetable cu intrari “None”.

La fiecare pas, pentru o stare “X”, genereaza starile vecine in felul urmator:

- ia, pe rand, toate intrarile cu valori diferite de “None” din timetable si le face, pe rand, “None”
- ia, pe rand, toate intrarile cu valori “None” si completeaza intrarea cu toate posibilitatile de intrari (*)

Practic, de la o stare “X”, starile vecine vor fi toate starile obtinute prin adugarea si stergerea unei singure intrari.

(*) Optimizare: nu se completeaza cu absolut toate posibilitatile de intrari, ci se respecta din start 2 restrictii hard pentru a nu mai fi nevoiti sa le verificam mai tarziu, pentru toate starile generate. Cele doua constrangeri sunt:

1) toti profesorii predau doar materiile pe care sunt specializati

2) in toate salile se tin ore doar la materiile pentru care este repartizata sala

Cand am verificat constrangerile pentru fiecare sala, am tinut cont de urmatoarele lucruri: o constrangere hard ar trebui sa aiba un cost mai mare decat toate constrangerile soft adunate, asa ca, pentru fiecare constrangere soft am pus costul 1, iar pentru fiecare constrangere hard am pus costul 10000, cu o singura exceptie:

Pentru conditia de acoperire, daca avem doua stari “X” si “Y”, “X” nu acopera niciun loc pentru o anumita materie, iar “Y” acopera partial, atunci nu este corect sa aiba ambele stari costul 10000, “Y” fiind o stare mai buna. Prin urmare, am decis ca acoperirea partiala ar trebui sa aiba un cost < 10000, dar totusi sa fie

mai mare decat suma tuturor costurilor constrangerilor soft care pot fi nerespectate: cost = 10000 - acoperire (cu cat este acoperirea mai mare, cu atat costul va fi mai mic)

2) Optimizari si trade-off-uri

Optimizarea algoritmului consta in faptul ca nu am construit toate starile posibile, ci le-am exclus din start pe cele redundante, care nu merita explorate, astfel salvand timp de executie.

Am incercat si o alta abordare a problemei (intr-un alt Jupyter Notebook), care presupunea calcularea costului total al restrictiilor pe baza starii anterioare, incercand sa salveze timp la calculul restrictiilor. Pe de-o parte castigam timp astfel, dar aveam nevoie sa tin informatii in dictionare pentru fiecare stare in parte. Astfel, fiecare stare avand dictionarele ei, eram nevoit sa fac “deepcopy” pentru toate informatiile atunci cand exploram o stare diferita, iar acest procedeu consuma mult timp, mai mult decat solutia aleasa in prezent.

Consider ca am ales un trade-off bun. Varianta de Hill Climbing aleasa este cea standard, care alege urmatoarea stare ca fiind cea mai buna. Timpul de rulare este unul acceptabil, chiar si pentru cel mai mare test, iar constrangerile incalcate pentru starea finala sunt, la fel, acceptabile. Daca as fi ales First Choice Hill Climbing, ar fi fost mai rapid, dar avea sanse mai mici sa dea o solutie decenta la sfarsit. Varianta Random Restart ar fi putut fi o optiune daca nu era nevoie sa exploreze atat de multe stari. Deja sunt explorate foarte multe stari pentru varianta aleasa, iar timpul ar fi fost prea mare pentru varianta Random Restart.

Mai jos sunt informatii legate de timpul de executie, numarul de stari construite si calitatea solutiei, pentru fiecare fisier de input:

---

dummy.yaml:

Timp: 0.03800605773925781

Cost conflicte: 0

Stari explorate: 666

---

orar_mic_exact.yaml:

Timp: 4.797006034851074

Cost conflicte: 9761

Stari explorate: 17205

Materia PL nu are acoperirea necesară! -> +9760

Profesorul Andrei Ilie nu dorește să predea în ziua Miercuri! -> +1

---

orar_bonus_exact.yaml:

Timp: 120.61576771736145

Cost conflicte: 10000

Stari explorate: 230634

Materia PM nu are acoperirea necesară! -> +10000

---

orar_mediu_relaxat.yaml:
Timp: 51.324047899246216

Cost conflicte: 0

Stari explorate: 131139

---

orar_constrans_incalcat.yaml:

Timp: 16.027996683120728

Cost conflicte: 9730

Stari explorate: 47895

Materia PCom nu are acoperirea necesară! -> +9730

---

orar_mare_relaxat.yaml:

Timp: 357.34263157844543

Cost conflicte: 0

Stari explorate: 348504
