Projektni zadatak iz kolegija

Heurističke metode optimizacija

Optimizacija zamjena grupa studenata

s tabu pretraživanjem

Danijel Butković , 0246041206

Dino Perić , 0318003179

# Opis problema

Na FER-u studenti koriste zamjenu grupa kako bi promijenili grupu u kojoj bi htjeli slušati određenu aktivnost na predmetu (npr. predavanja na Fizici). Za što uspješniju zamjenu grupa potrebno je napraviti algoritam koji će temeljem prikupljenih zahtjeva studenata napraviti što veći broj zamjena studenata.

# Opis primijenjenog algoritma tj. heuristike (prikaz rješenja, funkcija cilja/prikladnosti, način dobivanja početnog rješenja, kriterij zaustavljanja i veličina iteracije, elementi dizajna specifični za odabranu heuristiku…)

Algoritam koji je korišten za rješavanje problema je pretraga lokalnog susjedstva kombinirana s tabu listom. Rješenje prikazujemo s binarnim vektorom koji je jednake duljine kao i broj zahtjeva.

Lokalno susjedstvo generiramo tako da uzmemo pola nasumično odabranih indeksa unutar binarnog vektora te na njima radimo bitflip te provjeravamo može li se taj zahtjev odobriti ili će se odbiti. Uzimamo u obzir samo one susjede gdje je bitflip moguć.

Algoritam isprobava kombinacije te provjerava da li zahtjev zadovoljava ograničenja. Pomoću binarnog vektora koji je duljine kao i broj zahtjeva, radi se bitflip te pomoću tabu pretraživanja se ta pozicija ne mijenja sljedećih *n* iteracija. Taj binarni vektor predstavlja rješenje našeg problema.

Funkcija cilja je *Score* funkcija koja računa nagrade za napravljene zamjene te kazne ako je broj studenata u grupi manji od poželjnog minimalnog broja ili veći od poželjnog maksimalnog broja ljudi u grupi.

Početno rješenje se dobiva tako da uzmemo binarni vektor(u kojem su svi elementi 0) te nasumično odaberemo 1/10 (jedna desetina) zahtjeva te provjeravamo da li zadovoljavaju ograničenja te ih ažuriramo ako zadovoljavaju.

Kriterij zaustavljanja izvođenja programa je vremensko ograničenje. Kao argument pri pozivu programa se koristi parametar *timeout* koji je varijabla tipa *integer* te predstavlja vrijeme izvođenja programa u sekundama.

Elementi koji su specifični za odabranu heuristiku:

* Binarni vektor – predstavlja listu zahtjeva te sadržava samo 0 ili 1 na svakoj poziciji.

1 predstavlja da je zahtjev odobren, 0 predstavlja da je zahtjev odbijen.

* Tabu lista – koristi se za izbjegavanje lokalnog optimuma na način da bit koji se promjeni

u binarnom vektoru ne smije mijenjati sljedećih *n* iteracija

* Generiranje susjedstva – pretraživanje susjeda u nadi da ćemo naći bolje rješenj
* Broj generiranih susjeda

# Pseudokod primijenjenog algoritma

Generiraj početni vektor

najboljiSusjed = početniVektor

Dok ima vremena:

Generiraj susjedstvo za najboljiSusjed:

Odaberi nasumične indekse na kojima će se vršiti bit flip

Za indeks u vektoru:

Ako je moguće izvršiti zamjenu spremi susjeda

Odaberi najboljeg susjeda:

Za svakog susjeda Izračunaj ocjenu

Odaberi susjeda sa najboljom ocjenom

Dodaj index zamijenjenog bita u tabu listu

Kraj

# Opis dobivenih parametara i diskusija (npr. utjecaj određenih parametara heuristike na kvalitetu rješenja, vrijeme izvođenja algoritma)

Rezultati

Sample 3

Award activity: ['0', '2', '4']

Award student: 1

Minmax penalty: 1

Iterations: 498

Time elapsed: 600.5764336585999

Best score: -7546

Number of requests given: 697

Sample 3

Award activity: ['0', '2', '4']

Award student: 1

Minmax penalty: 1

Iterations: 1389

Time elapsed: 1800.8971588611603

Best score: -4008

Number of requests given: 1531

Sample 3

Award activity: ['0', '2', '4']

Award student: 1

Minmax penalty: 1

Iterations: 2330

Time elapsed: 3600.796347141266

Best score: -2991

Number of requests given: 1808

Sample 4

Award activity: ['0', '2', '4']

Award student: 1

Minmax penalty: 1

Iterations: 1781

Time elapsed: 600.2963931560516

Best score: -2253

Number of requests given: 907

Sample 4

Award activity: ['0', '2', '4']

Award student: 1

Minmax penalty: 1

Iterations: 5351

Time elapsed: 1800.2538838386536

Best score: -2200

Number of requests given: 891

Sample 4

Award activity: ['0', '2', '4']

Award student: 1

Minmax penalty: 1

Iterations: 10947

Time elapsed: 3600.0730998516083

Best score: -2211

Number of requests given: 899

Sample 5

Award activity: ['0', '2', '4']

Award student: 1

Minmax penalty: 1

Iterations: 29199

Time elapsed: 3600.059859275818

Best score: -43

Number of requests given: 526

Sample 5

Award activity: ['0', '2', '4']

Award student: 1

Minmax penalty: 1

Iterations: 14861

Time elapsed: 1800.046605348587

Best score: -9

Number of requests given: 530

Sample 5

Award activity: ['0', '2', '4']

Award student: 1

Minmax penalty: 1

Iterations: 5030

Time elapsed: 600.0781738758087

Best score: -3

Number of requests given: 548

Sample 6

Award activity: ['0', '2', '4']

Award student: 1

Minmax penalty: 2

Iterations: 18050

Time elapsed: 3600.1824045181274

Best score: -520

Number of requests given: 785

Sample 6

Award activity: ['0', '2', '4']

Award student: 1

Minmax penalty: 2

Iterations: 8995

Time elapsed: 1800.0812602043152

Best score: -518

Number of requests given: 787

Sample 6

Award activity: ['0', '2', '4']

Award student: 1

Minmax penalty: 2

Iterations: 2705

Time elapsed: 600.207649230957

Best score: -476

Number of requests given: 775

Sample 7

Award activity: ['0', '2', '4']

Award student: 1

Minmax penalty: 1

Iterations: 1438

Time elapsed: 600.4533398151398

Best score: -3798

Number of requests given: 1294

Sample 7

Award activity: ['0', '2', '4']

Award student: 1

Minmax penalty: 1

Iterations: 4083

Time elapsed: 1800.3956537246704

Best score: -3877

Number of requests given: 1287

Sample 7

Award activity: ['0', '2', '4']

Award student: 1

Minmax penalty: 1

Iterations: 8246

Time elapsed: 3600.1491680145264

Best score: -3935

Number of requests given: 1252

Iz gornjih podataka je vidljivo da vrijeme izvođenja utječe na krajnje rješenje na način da će maksimizirati funkciju cilja (Score) s većim brojem iteracija.

# Diskusiju o „fairness-u“ zamjene grupa (npr. treba li algoritam prioritizirati studente koji traže samo jednu zamjenu ili studente koji traže više zamjena; je li pošteno da studentu s traženom jednom zamjenom nije provedena zamjena, dok su studentu koji je tražio 5 zamjena, provedene 3 ili čak svih 5 zamjena; je li primjerice bolje, u slučaju ako se može provesti samo 1/5 zamjena, provesti tu zamjenu ili ne provoditi zamjene uopće). Također, u raznim slučajevima, prokomentirajte kako biste definirali nagrade/kazne (težinske faktore), s ciljem unaprjeđenja zamjene grupa na FER-u. Primjerice, osvrnite se na to kako bodovima nagraditi ili kazniti zamjenu grupa za studente koji su imali 1/1 zamjenu [broj\_provedenih\_zamjena/broj\_traženih\_zamjena], 0/1, 0/n, 1/n, 3/n, n/n)

Algoritam ne preferira niti jednog studenta te ne radi razliku između studenta s jednom zahtjevom ili studenta s 5 zahtjeva za zamjenom grupa. Zahtjevi se uzimaju slučajnim odabirom tako da svi imaju jednaku šansu da im se zahtjev odobri. To bi trebalo raditi fairness između odabira zahtjeva.