# 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 rješenje

od trenutnog.

# Pseudokod primijenjenog algoritma

Generiraj početno rješenje

Dok ima vremena:

Generiraj susjedstvo

Odaberi najboljeg susjeda

Dodaj susjeda u tabu listu

Kraj

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