Implementacija CSP na problemu raspoređivanja redosleda tačnih odgovora na testu sa ponuđenim odgovorima

Izveštaj

Aleksa Stančev 17434  
Miloš Jovanović 17158

**Sadržaj**

[1. Kratak opis problema 2](#__RefHeading___Toc385_553374464)

[2. Pregled i kratak opis tehnika i algoritama 2](#__RefHeading___Toc387_553374464)

[3. Formulacija problema 2](#__RefHeading___Toc389_553374464)

[4. Opis rešenja 3](#__RefHeading___Toc393_553374464)

# Kratak opis problema

Ovaj projekat za cilj ima automatizaciju procesa izbora određenog broja pitanja sa ponuđenim odgovorima iz unapred pripremljenog skupa pitanja, kao i raspoređivanja ponuđenih odgovora.

Problem koji rešavamo se sastoji iz više celina, u koje spadaju:

- izbor grupe pitanja iz skupa unapred poznatih pitanja

- generisanje od 1 do 4 (broj bira korisnik) kombinacije rasporeda pitanja

- raspoređivanje tačnog odgovora svakog pitanja na jednu od 4 opcija (A, B, C ili D)

# Pregled i kratak opis tehnika i algoritama

Za izradu aplikacije korišćen je Python progarmski jezik, bez dodatnih paketa.

Izbor pitanja koje dolaze u obzir za generisanje jednog testa se vrši korišćenjem choice i shuffle funkcija, uz primenu predhodno definisanih ograničenja.

Za raspoređivanje odgovora se korišti „Backtacking“ traženje u kombinaciji sa „Forward checking“ tehnikom i „Degree Heuristic“.

# Formulacija problema

Glavni fokus aplikacije je što ravnomernije raspoređivanje ponuđenih odgovora za izabrana pitanja. Raspoređivanje se vrši nasumično, ali su validne jedino kombinacije koje su u okviru sledećih ograničenja:

- pitanja sa istim rednim brojem, ali u različitim kombinacijama ne smeju imati isti raspored ponuđenih odgovora

- jedno pitanje poseduje jedinstven raspored ponuđenih odgovora za svaku kombinaciju u kojoj se nalazi

- susedna pitanja u jednoj kombinaciji nemaju tačan odgovor na istim pozicijama

Ovim ograničenjima je opisan CSP (Constraint Satisfaction Problem) koji ova aplikacija rešava.

# Opis rešenja

Za pravilan rad aplikacije neophodan je ulazni fajl sa unesenim pitanjima u tačno određenom formatu. Iz tog fajla se učitavaju potrebne informacije vezane za pitanja, dok ostale parametre korisnik unosi ručno.

Na osnovu učitanih vrednosti, primenom ograničenja za filtriranje i tehnika nasumičnog razmeštanja vrši se selekcija pitanja za svaki test pojedinačno (korisnik unosi koliko testova želi da generiše). Tako selektovana pitanja se dalje mapiraju u čvorove grafa, koji se kasnije koristi u gorenavedenim algoritmima. Dobijene kombinacije se zatim snimaju u tekstualne fajlove u odredišnom folderu, koji je takođe korisnik uneo kao parametar.