

Projektni zadatak br. 3

Predmet: Nelinearno programiranje i evolutivni algoritmi

Tema: Genetski algoritam, problem pravljenja rasporeda

Dat je skup događaja opisanih imenom i trajanjem (u pitanju su predavanja i vežbe na fakultetu). Takođe je dat skup učionica opisanih imenom. Potrebno je napraviti raspored predavanja i vežbi tako što će se svakom događaju dodeliti određeno vreme, dan i učionica, tako da su zadovoljeni sledeći uslovi:

- Između događaja u istoj učionici mora postojati pauza od najmanje 15 minuta
- Nastava može početi najranije u 7:00h
- Nastava se može završiti najkasnije u 19:00h
- Nastava se može izvoditi samo radnim ranima (ponedeljak, utorak, sreda, četvrtak, petak)

Cilj je napraviti onaj raspored u kom je prosečno vreme početka nastave za svaku učionicu što kasnije, a prosečno vreme završetka nastave što ranije. Formalno (matematički) posmatrano: Zabeležimo za svaku učionicu i za svaki dan vreme proteklo od 7:00h ujutru do početka prvog predavanja. Ako imamo n učionica i m radnih dana, ovih vrednosti će biti $n \cdot m$. Obeležimo ove vrednosti sa p_i , gde i uzima celobrojne vrednosti od 1 do $n \cdot m$. Na sličan način zabeležimo vreme proteklo od kraja poslednjeg predavanja do 19:00h za svaki dan i za svaku učionicu. Ova vremena obeležimo sa k_i . Kriterijum optimalnosti formiramo na sledeći način:

$$f(\mathbf{p}, \mathbf{k}) = \sum_{i=1}^{n \cdot m} p_i \cdot k_i$$

Traži se maksimum kriterijuma optimalnosti.

Implementirati rešavanje ovog problema koristeći genetski algoritam. Podaci su dati u datoteci `data_timetable.txt`. U prvom redu su navedene oznake učionica. Počev od trećeg reda pa nadalje, u svakom redu je dat jedan događaj opisan nazivom i trajanjem u minutima. Koristiti programski jezik python. Rešenje problema predstavlja raspored događaja za svaku učionicu za svaki dan. Nije neophodno koristiti grafički prikaz ili interfejs.

Dokumentovati projekat koristeći šablon za dokumentaciju dostupan na sajtu predmeta. Dokumentacija treba da sadrži između dve i pet strana A4 formata. U dokumentaciji je neophodno opisati:

- Strukturu programa
- Ograničenja problema
- Način implementacije operatora mutacije i ukrštanja
- Strategiju odabira jedinki za ukrštanje
- Odabir parametara algoritma
- Rezultate algoritma