**Univerzitet u Banjoj Luci**

**Elektrotehnički fakultet**

**Katedra za automatiku**

**Metodi vještačke inteligencije**

Izvještaj o urađenom projektnom zadatku

**Genetički algoritmi**

Student:

Halim Bačić, 1330/14

# Opis problema

Korištenjem genetičkog algoritma, određujemo minimum funkcije:

i to na intervalu gdje

U drugom dijelu projekta, potrebno je odrediti maksimum presjeka date funkcije i ravni nastele presjekom x, y, z ose za x=0.

# Rješenje

Do rješenja se dolazi korištenjem programa pisanim u Java programskom jeziku. Bitne klase koje su korištene u rješavanju problema su: Point3D, Point2D, Population i Simulation.

Point3D i Point2D su klase koje nasljeđuju klasu Point. Point3D se koristi u prvom djelu zadatka, a Point2D se koristi u rješavanju drugog dijela zadatka. Klasa Point sadrži atribut *fitness* koji se koristi u selekciji rješenja i selekciji jedinki populacije pri korištenju genetičkog algoritma. Metoda compareTo je implementirana tako da se pri poređenju objekata tipa Point2D i Point3D vrši poređenje po vrijednosti atributa *fitness.*

Klasa Population sadrži dva atributa, a to su trenutna populacija i redni broj populacije. Bitne metode koje se nalaze u klasi Population su generatePopulationPoint3D I generatePopulationPoint2D koje se koriste za inicijalizovanje početnih populacija. Početnu populaciju čine slučajno generisane koordinate tački. Vrijednosti atributa se nalaze u pomenutim intervalima za svaku od generisanih tačaka. Klasa Population sadrži metodu sortPopulation koja sortira jedinke u populaciji na osnovu fitness vrijednosti, te se na taj način biraju elementi koji će se prenositi u narednu generaciju.

Klasa Simulation je glavna klasa zadužena za implementaciju algoritma. Atributi klase su vrijednosti koje su u praksi ključne za implementaciju algoritma, a to su veličina populacije, broj iteracija (koji su cjelobrojne vrijednosti) te vjerovatnoće za ukrštanje jedinki i mutaciju novih jedinki. Klasa Simulation sadrži metode kojima se vrši sortiranje populacije, izračunavanje fitness vrijednosti, funkcije za ukrštanje te funkcije za mutaciju novih jedinki. Klasa Simulation implementira interfejs BinaryRepresentation koji sadrži metode za konverziju Double vrijednost u binarni oblik i obrnuto da bi se održao koncept genetičkog algoritma.

**1. Generisanje inicijalne populacije**

Nakon unosa vrijednosti za ….

Uočiti kako varijacije pojedinih parametara utiču na kvalitet rješenja. Pokušati doći do optimalnih parametara koji za najkraće vrijeme daju zadovoljavajuće rezultate.

Potrebno je izvršiti minimalno 10 eksperimenata. Rezultate prikazati na pogodan način: tabelarno, grafički i sl.

Nije potrebno objašnjavati programski kod nego algoritam pomoću koga se došlo do rješenja, odnosno osnovnu ideju!

# Primjeri formatiranja

U ovom Poglavlju je dat način referenciranja tabela, slika, listinga koda, matematičkih formula, kao i sami prikazi istih. Numeracija je formata *BrojPoglavlja.RedniBrojPrikazaUPoglavlju*.

Primjer tabelarnog prikaza dat je u Tabeli 4.1.

Tabela 4.1 *Primjer tabelarnog prikaza*

|  |  |  |
| --- | --- | --- |
|  | **Atribut 1** | **Atribut 2** |
| **Varijanta 1** | Rezultat 1.1 | Rezultat 1.2 |
| **Varijanta 2** | Rezultat 2.1 | Rezultat 2.2 |

Dozvoljene su varijacije izgleda, ali je neophodno da izgled tabela bude unificiran u cijelom izvještaju. Primjetiti da se numeracija i naziv tabele postavljaju **iznad** tabele sa lijevim poravnanjem.

Primjer prikaza slikom dat je na Slici 4.1.



Slika 4.1 *Primjer prikaza slikom*

Primjetiti da se numeracija i naziv slike postavljaju **ispod** slike, pri čemu i slika i naziv imaju **srednje** poravnanje.

Primjer listinga sa kodom dat je u Listingu 4.1.

Listing 4.1 *Primjer prikaza koda*

|  |
| --- |
| #include <stdio.h>  int main(int argc, char \*argv[])  {  printf(“Hello, world!\n”);  return 0;  } |

Nije neophodno navođenje listinga sa programskim kodom za svaku funkciju, proceduru itd. Ovdje je dat samo primjer prikaza programskog koda, ukoliko ima potrebe da se isti iz nekog razloga prikaže u izvještaju. Primjetiti da se naziv i numeracija listinga pišu **iznad** listinga sa **desnim** poravnanjem.

Primjer matematičkog zapisa formule dat je sa (4.1).

(4.1)

Primjetiti da jednačina ima svoju numeraciju koja je obavezna, te da ista ima desno poravnanje.

Zadržati font, veličinu, formatiranje paragrafa i formatiranje papira.

Ovo poglavlje služi isključivo za ilustraciju prikaza slika, tabela i jednačina te ne treba kao takvo da se nalazi u izvještaju projektnog zadatka.

# Zaključak

Unutar zaključka potrebno je dati vlastito mišljenje o konkretnom algoritmu, prednostima i nedostacima istog koji su uočeni tokom izrade projektnog zadatka. Za koje parametre algoritam daje najbolje rješenje i zašto? Da li postoji način da se algoritam unaprijedi i poboljša? Koje su praktične koristi analiziranog algoritma i sl.? Voditi računa da se izvještaj nikada ne piše u prvom licu! Dakle, rečenice poput: “Ja mislim…”, “Zaključio(la) sam…” i slične su suvišne i kao takve ne treba da se nalaze u izveštaju. Pravilna formulacija zahtijeva treće lice, npr.: “U radu se došlo do zaključka…”, “Zaključili smo…” i sl.