**Univerzitet u Banjoj Luci**

**Elektrotehnički fakultet**

**Katedra za automatiku**

**Metodi vještačke inteligencije**

Izvještaj o urađenom projektnom zadatku

**Genetički algoritmi**

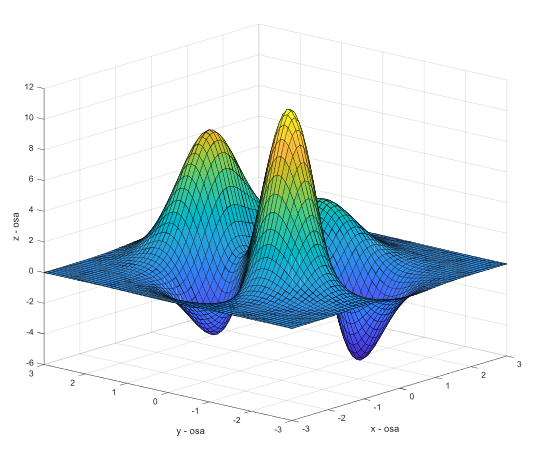
Student:

Dragan Kos, 1233/15

# Opis problema

Odrediti globalni minimum i globalni maksimum funkcije z(x,y) na intervalu na

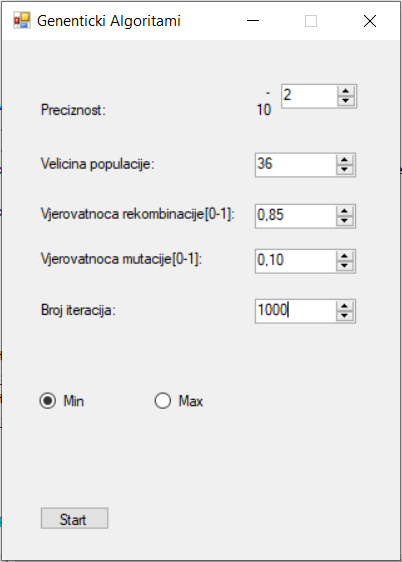
kome je funkcija prikazana na Slici 1. Analitički izraz funkcije dat je sa (1.1).

**

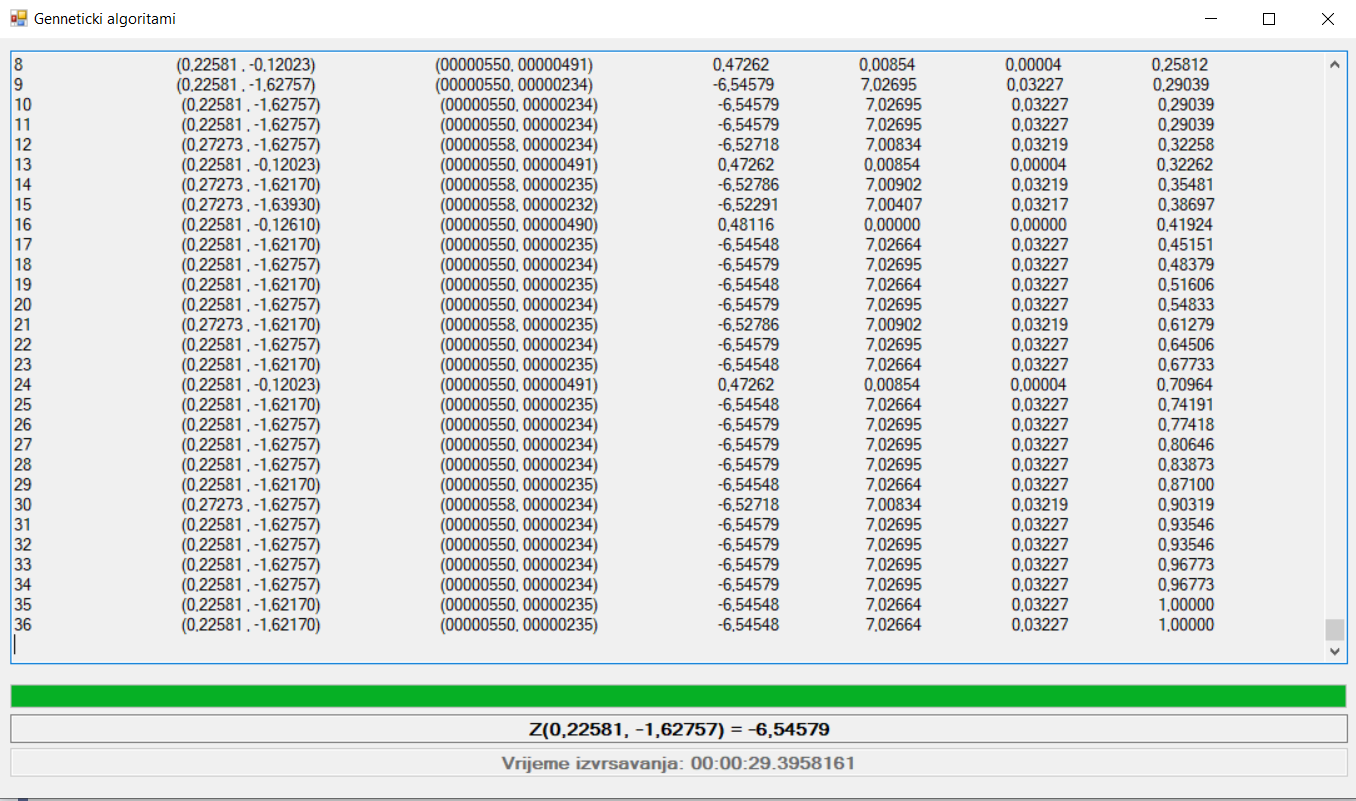
*Slika 1. – Grafik funkcije z(x,y)*

# Rješenje

Rješenje zadatka je realizovano kao desktop aplikacija napisana u C# programskom jeziku koja implementira geneticki algoritam za rješavanje opisanog problema. Aplikacija omogucava unos parametara (velicina populacije, vjerovatnoća rekombinacije i mutacije, broj iteracija, preciznost te da li ce se vršiti izračunavanje globalnog minimuma i globalnog maksimuma) kroz odgovarajući GUI(Graficki korisnički interfejs) prikazan na Slici 2. Klikom na dugme Start pokreće se glavni program za izračunavanje i otvara se nova forma na kojoj se ispisuju međugeneracije kao i konačno rješenje(Slika 3).



Slika 2. – Grafiči korisnički interfejs za unos parametara



Slika 3. – Nova forma za prikaz rezultata

Aplikacija sadrži sljedeće klase:

* Point – predstavlja tačku koja sadrži x, y koordinatu, vrijednost funkcije z u toj tački, vrijednost fintes funkcije(FF), vjerovatnocu izbora tačke za sljedeću generaciju(p), kumulativnu vjerovatnoću(q), kodovane rijednosti x i y koordinate(CodedX, CodedY), broj bita potrebnih za kodovanje(n) i bitsku reprezentaciju (xBits i yBits). Ova klasa sadrži i funkcije za:
  + kodovanje,
  + dekodovanje,
  + konverziju bita u cijeli broj i obrnuto,
  + realizaciju mutacije
  + realizaciju rekombinacije(ukrštanja).
* GenAlg – predstavlja klasu u kojoj je implementiran genetički algoritam. U ovoj klasi su definisane promjenljive:
  + velicina popilacije(size),
  + broj iteracija(iterations),
  + vjerovatnoća rekombinacije(probCo),
  + vjerovatnoća mutacije(probMut),
  + fitnes funkcija(ff).

Klasa GenAlg sadrži sljedeće funkcije:

* Start – u ovoj funkciji je implementiran genetički algoritam,
* Map – vrši mapiranje slućajnog broja sa intervala [0,1] na interval [-3,3],
* Calculate – pomoćna funkcija u kojoj se vrši izračunavanje ocjena jedinki i njihovih vjerovatnoca,
* FFMin – fitnes funkcija za traženje minimuma,
* FFMax – fitnes funkcija za traženje maksimuma.
* Ostale klase nisu od velikog značaja jer se one odnose na GUI.

U klasi GenAlg se nalazi funkcija Start u kojoj se na pseudoslučajan način generiše potreban broj jedinki za početnu populaciju. Prilikom generisanja svake tačke odmah se izarčunava vrijednost funkcije z u toj tački, vrši se njeno kodovanje i računa se binarni oblik svake koordinate.

Kodovanje se vrši tako sto se svaka koordinata koduje na osnovu formule (2.1) gdje je:

* Gd - donja granica intervala,
* Gg - gornja granica intervala i
* n - potreban broj bita za kodovanje.

Potreban broj bita se određuje formulom (2.2) gdje p prestavlja izbranu preciznost.

Dekodovanje svake kordinate se vrši na pomoću formule (2.3).

Zatim se bira fitnes funkcija(u zavisnosti da li se traži minimum ili maksimum) i vrši ocjenjivanje svake jedinke u populaciji. Fitnes funkcija za traženje globalnog minimuma data je sa (2.4), a za traženje globalnog maksimuma sa (2.5). Ove fitnes funkcije su izabrane zato što daju nenegativne vrijednosti, najveću vrijednost za najbolje prilagođenu jedinku a nultu vrijednost za najlošije prilagođenu.

Da bi se izračunala vjerovatnoca izbora prvo se mora izračunati ukupna ocjena populacije. Ukupna ocjena populacije se računa kao suma vrijednosti fintnes funkcije svih jedinki (2.6).

(2.6)

Na osnovu izračunate ocjene populacije računa se vjerovatnoča izbora za svaku jedinku tako što se vrijednost fitnes finkcije te jedinke potijeli sa ukupnom ocjenom populacije (2.7).

Za tako izračunatu vjerovantnoću izbora računa se kumulativna vjerovatnoća prema (2.8) ili prema (2.9).

U zavisnosti od kumulativne vejrovatnoće i pseudoslučajno generisanog broja r sa intervala [0,1] vrši se izbor jedinki za sljedeću generaciju(ruletska selekcija). Ako je data se uzima jedinka sa rednim brojem nula ili ako je tada se bira jedinka rednog broja i.

Osim ruletske selekcije korisi se još i elitistička selekcija. Elististička selekcija podrazumjeva da najbolje ocjenjena jedinka direktno prolazi u sljedeću generaciju.

Neke od izabranih jedinki će proći nepromijenjene u sledeću genraciju, a neke će se ukrstiti. Ukrštanje se realizuje tako što se izabrane jedinke izmiješaju dovoljan broj puta i onda se provjerava da li će se vršiti ukrštanje dvije susjedne jedinke. Broj miješanja se generiše tako što se uzme polovina veličine populacije i doda se pseudoslučajan broj sa intervala [0,velicina populacije], zatim se na pseudoslučajan način biraju parovi koji će se mijenjati. Zavisno od izabrane vjerovatnoće ukštanja(probCo) vršiće se ukrštanje ako je ispunjen uslov da je .

Ukrštanje se vrši u jednoj tački tako što se na pseudoslučajan način generiše tačka ukrštanja(indeks do koga će se mijenjati biti). Jedinka se posmatra na sljedeci način:

.

Neka se ukrštanje dvije jedinke vrši u tački 3 tada se dobija:

Posljednji korak u jednoj iteraciji genetičkog algoritma je mutacija. Mutacija prestavlja slučajnu promjenu jednog gena(bita) jedinke. Mutacija sprečava da se populacija skoncentriše oko lokalnog minimuma(maksimuma) tako sto se kod nekih hromozoma(jedinki) vrši promjena jednog gena(bita) i tako nastane hromozom koji se moze približiti globalnom minimumu(maksimumu). Mutacija se vrši ako je pseudoslučajni broj r manji od izabrane vejrovatnove mutacije(probMut).

Nakon mutacije dobija se nova populacija. Za ovako nastalu novu populaciju računaju se ocjene i vjerovatnoće i algoritam se izvršava ciklično. Algoritam se završava kada se dostigne uneseni broj iteracija ili kada sve jednike u populaciju postanu jednake(imaju iste koordinate). Ako su sve jedinke u populaciji jednake onda je potrebno zaustaviti algoritam jer se ne može formirati nova generacija zato što sve jedinke imaju lošu ocjenu(vrijednost fitnes funkcije je nula).

# Analiza rada programa

U ovom poglavlju biće prikazani rezultati testova u zasnosti od različitih parametara. Kako je na laboratoriskim vježbama korištena populacije veličine 8 jedinki, sa tom istom veličinom ćemo početi ovdje. U sljedecim tabelama su prikazane srednje vrijednosti rezultata i srednje vrijeme izvršavanja. Za svaku kombinaciju parametara izvršeno je 10 mjerenja.

Tabela 3.1 *Prikaz rezultata globalnog minimuma za veličinu populacije od 8 jedinki*

|  |  |  |  |
| --- | --- | --- | --- |
| **Vjerovatnoća ukrštanja** | **Vjerovatnoća mutacije** | **Srednja vrijednost minimuma** | **Srednje vrijeme izvršavanja [s]** |
| 0,30 | 0,70 | -1,20103 | 1,31 |
| 0,50 | 0,25 | -3,71912 | 0,71 |
| 0,50 | 0,10 | -1,89912 | 0,21 |
| 0.85 | 0,25 | -4,54757‬ | 0,37 |
| 0,85 | 0,10 | -3,37047 | 0,43 |

Tabela 3.2 *Prikaz rezultata globalnog maksimuma za veličinu populacije od 8 jedinki*

|  |  |  |  |
| --- | --- | --- | --- |
| **Vjerovatnoća ukrštanja** | **Vjerovatnoća mutacije** | **Srednja vrijednost maksimuma** | **Srednje vrijeme izvršavanja [s]** |
| 0,30 | 0,70 | 2,68650 | 1,56 |
| 0,50 | 0,25 | 4,72311 | 0,22 |
| 0,50 | 0,10 | 4,24418 | 0,37 |
| 0.85 | 0,25 | 5,49390 | 0,74 |
| 0,85 | 0,10 | 4,63162 | 0,24 |

Tabela 3.3 *Prikaz* *rezultata globalnog minimuma* *za* *veličinu* *populacije* *od* *16* *jedinki*

|  |  |  |  |
| --- | --- | --- | --- |
| **Vjerovatnoća ukrštanja** | **Vjerovatnoća mutacije** | **Srednja vrijednost minimuma** | **Srednje vrijeme izvršavanja [s]** |
| 0,30 | 0,70 | -5,17649 | 22,53 |
| 0,50 | 0,25 | -5,29614 | 1,29 |
| 0,50 | 0,10 | -6,01790‬ | 0,52 |
| 0.85 | 0,25 | -5,21097‬ | 0,25 |
| 0,85 | 0,10 | -5,77781 | 0,52 |

Tabela 3.4 Prikaz rezultata globalnog maksimuma za veličinu populacije od 16 jedinki

|  |  |  |  |
| --- | --- | --- | --- |
| **Vjerovatnoća ukrštanja** | **Vjerovatnoća mutacije** | **Srednja vrijednost maksimuma** | **Srednje vrijeme izvršavanja [s]** |
| 0,30 | 0,70 | 6,96622‬ | 24,17 |
| 0,50 | 0,25 | 7,11936‬ | 23,21 |
| 0,50 | 0,10 | 7,47855 | 23,03 |
| 0.85 | 0,25 | 8,07447 | 20,42 |
| 0,85 | 0,10 | 7,03273 | 18,19 |

Tabela 3.5 *Prikaz rezultata globalnog minimuma za veličinu populacije od 36 jedinki*

|  |  |  |  |
| --- | --- | --- | --- |
| **Vjerovatnoća ukrštanja** | **Vjerovatnoća mutacije** | **Srednja vrijednost minimuma** | **Srednje vrijeme izvršavanja [s]** |
| 0,30 | 0,70 | -5,68685 | 34,68 |
| 0,50 | 0,25 | -5,69156 | 30,32 |
| 0,50 | 0,10 | -6,20244 | 24,24 |
| 0.85 | 0,25 | -6,45550 | 30,30 |
| 0,85 | 0,10 | -6,45366 | 26,53 |

Tabela 3.6 *Prikaz rezultata globalnog maksimuma za veličinu populacije od 36 jedinki*

|  |  |  |  |
| --- | --- | --- | --- |
| **Vjerovatnoća ukrštanja** | **Vjerovatnoća mutacije** | **Srednja vrijednost maksimuma** | **Srednje vrijeme izvršavanja [s]** |
| 0,30 | 0,70 | 7,86475 | 34,80 |
| 0,50 | 0,25 | 8,03426 | 32,42 |
| 0,50 | 0,10 | 8,08152 | 30,79 |
| 0.85 | 0,25 | 8.08546 | 30,65 |
| 0,85 | 0,10 | 8,05584‬ | 28,82 |

Tabela 3.7 *Prikaz rezultata globalnog minimuma za veličinu populacije od 200 jedinki*

|  |  |  |  |
| --- | --- | --- | --- |
| **Vjerovatnoća ukrštanja** | **Vjerovatnoća mutacije** | **Srednja vrijednost minimuma** | **Srednje vrijeme izvršavanja [s]** |
| 0,30 | 0,70 | -6,53106 | 123,98 |
| 0,50 | 0,25 | -6,54560 | 99,24 |
| 0,50 | 0,10 | -6,47397‬ | 96,74 |
| 0.85 | 0,25 | -6,39877‬ | 97,27 |
| 0,85 | 0,10 | -6,54579 | 99,34 |

Tabela 3.8 *Prikaz rezultata globalnog maksimuma za veličinu populacije od 200 jedinki*

|  |  |  |  |
| --- | --- | --- | --- |
| **Vjerovatnoća ukrštanja** | **Vjerovatnoća mutacije** | **Srednja vrijednost maksimuma** | **Srednje vrijeme izvršavanja [s]** |
| 0,30 | 0,70 | 8,09706‬ | 115,04 |
| 0,50 | 0,25 | 8,09179 | 94,35 |
| 0,50 | 0,10 | 8,08106 | 96,96 |
| 0.85 | 0,25 | 8,08050‬ | 101,70 |
| 0,85 | 0,10 | 8,11558‬ | 98,30 |

Iz prethodnih tabela vidi se da za manje velicine populacije najbolja rješenja se dobijaju kada je vjerovatnoća mutacije relativno velika(oko 25%) jer pocetna populacija rijetko sadrži jedinke koje se nalaze u blizini globalnog minimuma/maksimuma. Zbog toga je potrebno da se mutacija češće vrši da se populacija izbaci iz ravnotežnog stanja, tj. da se neke jedinke udalje od ostatka populacije kako bi se izbjeglo da se populacija skoncentriše oko lokalnog minimma(maksimuma). Takođe se može vidjeti da se algoritam brže izvršava ako je vjerovatnoća ukrštanja i mutacije mala jer tada rijetko dolazi do promjena populacije i vrlo brzo sve jednike u populaciji postaju iste.

Za veliku populaciju poželjno je da vjerovatnoća mutacije bude manja(do 15%) jer je velika vjerovatnoća da početna populacija sadrži jedinke u blizini globalnog minimuma/maksimuma i zbog toga je potrebno spriječiti da se populacije previse udalji od tih jedinki. Što se tiče vjerovatnoće ukrštanja(rekombinacije) poželjno je da ona bude veća jer je velika vjerovatnoća da će se ukrštanjem dobro prilagođenih jedinki dobiti dobro prilagođeni potomci.

# Zaključak

Genetički algoritam daje skup rješenja čija je veličine jednaka veličini populacije. Najbolje prilagođeno rješenje iz tog skupa bi trebalo biti blizu stvarnog. Koliko blizu će ovo rješenje biti stavrnom zavisi od izabranih parametara.

Za vjrevatnoću rekombinacije od 80-90%, vjerovatnocu mutacije od 10-25% i veličinu populacije od 36 jedinki dobijaju se zadovoljavajuća rješenja za oko 30 sekundi što je za oko 3 puta brže nego kada je velićina populacije 200.