Univerzitet u Novom Sadu

Fakultet tehničkih nauka

Dokumentacija za projektni zadatak

Studenti: Rodušek Vladimir, SW23/2017

Pavlov Milan, SW35/2017

Predmet: Nelinearno programiranje i evolutivni algoritmi

Broj projektnog zadatka: 12

Tema projektnog zadatka: PSO algoritam, "black-box" optimizacija

# Opis problema

Data je funkcija f(**w**) koju smatramo nepoznatom. Argument funkcije **w** je vektorska veličina sa 60 dimenzija. Funkcija predstavlja veštačku neuronsku mrežu i nepoznata je u analitičkom smislu. Rešenje problema je pronalazak optimuma te funkcije koje predstavlja najmanju grešku same mreže.

Za optimizaciju funkcije koristiti PSO algoritam (Optimizacija rojem čestica).

Opis algoritma

PSO algoritam se oslanja na roj čestica za koji svaka čestica je opisana svojom pozicijom (**w** vektorska veličina sa **n** dimenzijom), vrednošću funcije u toj poziciji, njenom brzinom takođe definisanom u datoj **n** dimenziji i pored toga sadrži poziciju najbolje vrednosti funkcije (najbliža pozicija, te čestice, minimumu).

On za definisan broj iteracija pomera čestice, tj. čestice “zuje” oslanjajući se na pomeranje ka najboljoj poziciji koji je roj imao. U svakoj iteraciji se ažurira najbolja pozicija i same čestice, ali i roja tj. najbolja pozicija od svih čestica. Vektor brzine čestice se prema formuli pomera ka najboljem trenutnom rešenju.

Formula izleda ovako:

*i* – predstavlja trenutnu iteraciju

v – predstavlja brzinu

ω – predstavlja faktor inercije

cp – kognitivni faktor

cg – socijalni faktor

rp, rg – slučajni brojevi u intervalu (0, 1)

p – najbolja pozicija čestice

x – trenutna pozicija čestice

g – najbolja pozicija od svih čestica

inercijalna komponenta

kognitivna komponenta

socijalna komponenta

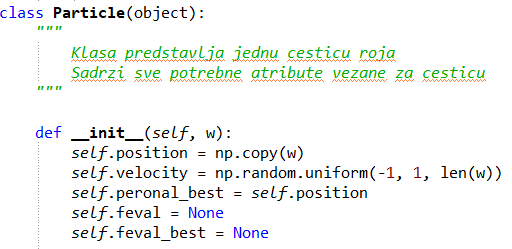
Opis faktora

* Uočeno je da se performanse algoritma (naročito preciznost u okolini rešenja) bitno poboljšavaju uvođenjem promenljivog inercionog faktora. Korišćeno je da se **ω** smanjuje od 0.9 do 0.4
* Faktor **cp** se tokom preterage smanjuje; u početku se jedinke vode sopstvenim iskustvom. Korišćeno je da se **cp** smanjuje od 2.5 do 0.5
* Faktor **cg** se u toku pretrage povećava; uticaj grupnog iskustva Korišćeno je da se **cg** povećava od 0.5 do 2.5

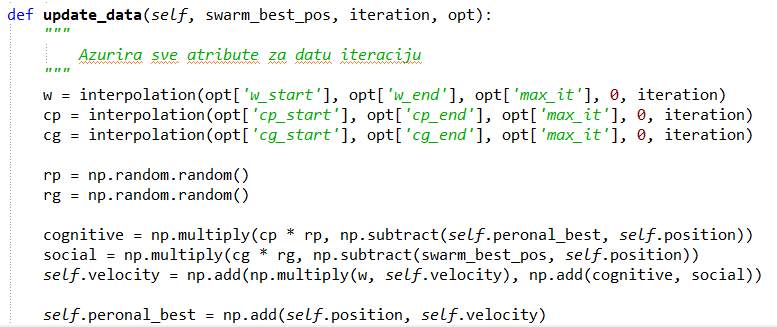
Struktura programa

Algoritam je implementiran u **Python3** programskom jeziku, za lakši rad sa matricama i nizovima korišćena je biblioteka **numpy**.

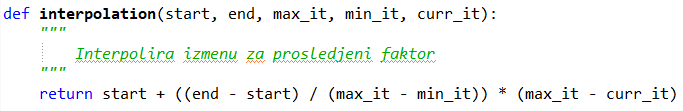
Čestica je realizovana kao klasa sa adekvatnim poljima i metodama.



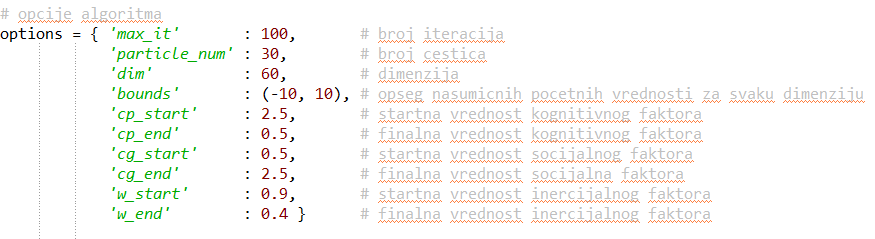
Prilikom svake iteracije za svaku česticu se pozivaju metode “evalute” i “update\_data”.



Za promenu faktora tokom petlji se koristi interpolacija prema datoj iteraciji



Za lakše biranje parametara algoritma realizovan je rečnik koji predstavlja opcije algoritma



Prilikom traženja optimalnog rešenja menjani parametri su broj iteracija, broj čestica kao i granice vrednosti za svaku od dimenzija čestice.

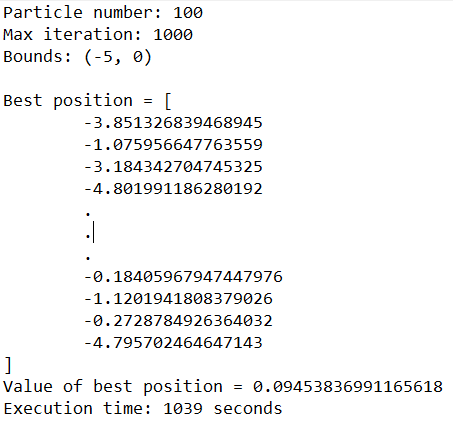
Rezultati algoritma

PSO algoritam ne može da garantuje da li je optimum samo lokalni ili je i globalni. Zbog nepoznavanja funkcije ne možemo konkretno odrediti, a ni proveriti tačnost rešenja, ali možemo da za različite faktore diskutujemo neka rešenja.

Za početno testiranje smo suzili intervale granice na dužinu od 5 tj. svaki interval se pomerao za 5. Time smo tražili lokalne minimume za podintervale intervala od -100 do 100. Najbolje rešenje nam je bilo na intervalu [-5, 0] i tu smo dobili da je greška vrednosti **0.11527677194021697**. Za svaki podinterval je korišćeno 100 čestica i 100 iteracija.

Primetivši da je interval [-5, 0] najbolji, fokusirali smo se na taj interval za dalje testiranje.

Najbolje rešenje je bilo za date parametre (broj čestica i broj iteracija):



**Napomena!** Odsečen je deo pozicije jer je prevelika slika za 60 dimenzija.

U log datoteci je u celokupnom stanju.