



Univerzitet u Novom Sadu

Fakultet tehničkih nauka

Dokumentacija za projektni zadatak

Januar 2021.

Studenti:

Ilija Kalinić, SW65/2019

Zoran Bukorac, SW40/2019

Matija Zarić, SW24/2019

Predmet:

Nelinearno Programiranje i Evolutivni Algoritmi

Broj projektnog zadatka:

br. 12

Tema projektnog zadatka:

PSO algoritam, "black-box" optimizacija

1 Opis problema

Zadata je funkcija $f(\underline{w})$ za koja je definisana preko veštačke neuronske mreže, pa ju je možemo smatrati nepoznatom. Poznato je da je \underline{w} vektor dimenzije 60, dakle:

$$\underline{w} = (w_1, w_2, w_3, \dots, w_{58}, w_{59}, w_{60}), \quad (1)$$

i da je vrednost kriterijuma realan, jednodimenzionalan broj.

Funkcija $f(\underline{w})$ je nepoznata u analitičkom smislu (ne postoji matematička formula koja opisuje tu funkciju), ali možemo evaluirati njenu vrednost za proizvoljnu vrednost argumenata \underline{w} . U priloženoj datoteci *ann_criterion.py* se nalazi kod za veštačku neuronsku mrežu odnosno kod za izračunavanje vrednosti našeg kriterijuma optimalnosti.

Potrebno je optimizovati ulazne vrednosti koristeći PSO (Particle Swarm Optimization) algoritam kako bismo minimizovali vrednost zadate nepoznate funkcije, tj. kriterijuma optimalnosti $f(\underline{w})$.

2 Uvod

2.1 Kratak opis Particle Swarm Optimization algoritma

Particle Swarm Optimization algoritam je genetički algoritam koji donekle simulira socijalna ponasanja zivotinja (najvise ptica) u velikim grupama. Glavni cilj algoritma PSO je da optimizuje zadat prostor pretrage koristeci roj ćestica u tom prostoru. Roj ćestica se može još i nazvati populacijom. Svaku ćesticu karakteriše njen položaj u prostoru, koji je ujedno i jedno potencialno rešenje problema. Inicijalne pozicije svih ćestica u prostoru se određuju nasumično. Ćestice određuju poziciju u sledećoj iteraciji na osnovu trenutne vrednosti kriterijuma optimalnosti (sopstvene brzine), sopstvene najoptimalnije vrednosti, najoptimalnije vrednosti u celoj populaciji. Uticaj sopstvene najoptimalnije vrednosti treba da bude veći u početnim iteracijama (ćestica sama traži optimum na osnovu svojih "najboljih iskustava"), a kako teče optimizacija da slabi, a uticaj najoptimalnije vrednosti u celoj populaciji treba da bude obrnut, manji u početku a veći na kraju optimizacije (ćestica se posle pridružuje ostatku populacije u pronalaženju minimuma).

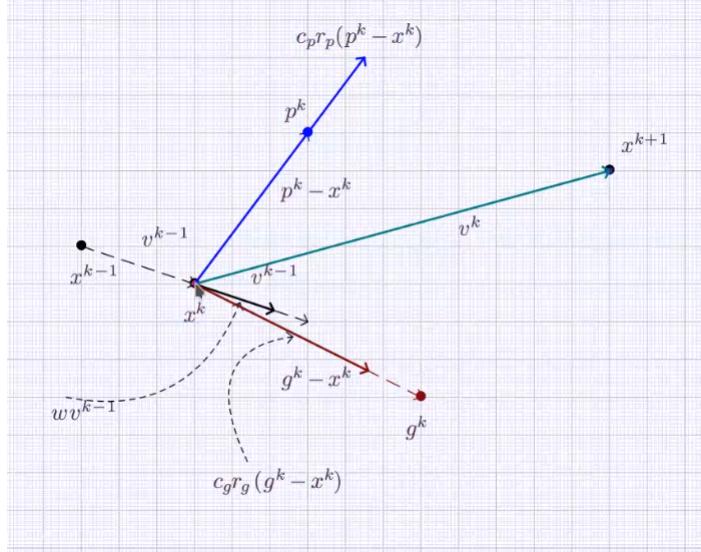


Figure 1: Skica kretanja ćestice u postupku optimizacije

3 Implementacija

Implementacija je odrađena u programskom jeziku **Python**, uz dve ugrađene biblioteke *math* i *random* i dodatnom bibliotekom za numeričke operacije *numpy*. U implementaciji se koriste 3 datoteke koji sadrže potrebne klase i kod za algoritam (*options.py*, *particle.py*

i *pso.py*) , priložena datoteka koja sadrži kriterijum optimalnosti (*ann_criterion.py* i skripta za pokretanje programa (*main.py*).

Implementirane su klase:

- **MyOptions** - sadrži atribute za sve opcije i parametre koji su potrebni za izvršavanje optimizacionog algoritma
- **Particle** - sadrži atribute u kojima se cuvaju vrednosti tacaka u toku cele optimizacije, brzine cestice u toku cele iteracije, tacka u kojoj je cestica bila najoptimalnija, kao i vrednost kriterijuma optimalnosti u toj tacki i broj dimenzija prostora. Dodatno postoje i funkcije za evaluiraće trenutne tacke u kojoj se nalazi cestica, i pomocne funkcije koje azuriraju ostale atribute.
- **PSO** - sadrži glavnu petlju za izvršavanje optimizacionog algoritma, a u atributima čuva vrednosti najbolje tacke i vrednosti k.o. u toj tački, populaciju (skup svih cestica) u listi i atribute za maksimalan broj iteracija i brojnost populacije.

4 Zaključak