February 11, 2015

Abstract

otvorit server u browseru nek ceka za kasnije

1 slajd

Dobar dan, mi smo bla i ble. Danas ćemo Vam predstaviti naš seminarski rad koji je usporedba nekoliko algoritama za traženje lokalnih ekstrema funkcija.

2 slajd

Ovo je struktura kojom ćemo probat pobliže objasniti sto smo i kako radili, te na kraju i demonstrirati našu aplikaciju koja pruža usporedbu tih algoritama.

3 slajd

piše isto na slajdu, to pokušamo zapamtit a ne da sve čitamo baš sa slajdova :)
Dvije osnovne vrste problema:

- traženje optimuma na segmentu domene (npr. Schafferova F6 funkcija na $[-100, 100]^2$)
- traženje optimuma funkcija ograničenih sa nekim jednakostima i/ili nejednakostima na cijeloj domeni (npr. G funkcije).

4 slajd

Dane problme rješavamo metaheuristikama. One se dijle na sljedeći način: *(pokazat na slide i pročitat šta piše).*
Mi ćemo se baviti ovim dijelom koji lokalno pretražuje prostor rješenja i daje nam jedno riješenje kao rezultat.

5 slajd

Algoritmi koje smo odabrali spadaju u klasu populacijskih algoritama, posebno u klasu algoritama baziranih na rojevim čestica. Ova vrsta algoritama se oslanja na kolektivno ponašanje samostalnih čestica u roju. Primjer su mravlji i pčelinji algoritam.

6 slajd

Sada ćemo dati kratki pregled znasvene i stručne literature u tom području unazad par godina.
Pročitat godinu, koji je algoritam u igri, možda pokoju funkciju

7 slajd

Pročitat godinu, koji je algoritam u igri, možda pokoju funkciju

8 slajd

Pročitat godinu, koji je algoritam u igri, možda pokoju funkciju

9 slajd

Pročitat godinu, koji je algoritam u igri, možda pokoju funkciju

10 slajd

Pročitat godinu, koji je algoritam u igri, možda pokoju funkciju
Ovaj članak je ustvari bio i motivacija za ovaj projekt, i vidimo da je nedavno izdan te da je tema jako popularna u znastvenoj zajednici.

11 slajd

No, što je zaista bio naš projektni zadatak?
Mi smo odabrali 2 algoritma koja se baziraju na roju čestica, PSO i ABC o kojima čemo malo kasnije, te smo osmisli novi algoritam koji je mini hibrid ta dva.
Za prikaz rezultata našeg projekta i interakciju s korisnikom osmislili smo i izradili grafičko web sučelje.

12 slajd

Evo kratkog pregleda korištenih algoritama sa kratkim opisom načina kako oni rade.
Prvi lagoritam je pčelinji algoritam (dalje čemo ga zvati ABC). Njegove mane su veliki broj parametara koje koristi *(pokazat rukom)* i to što ne konvergira. Prednost mu je što po samoj prirodi svog rada ima neku vrstu lokalne pretrage a veliki brj čestica koje sudjeluju u izračunu povećavaju šansu pronalska povoljnog rezultata.

13 slajd

pročitaj slajd, maši rukama dok objašnjavas, crtaj malo po ploči

14 slajd

Zatim, implementirali smo i PSO
pročitaj slajd točke 2 i 3

15 slajd

pročitaj slajd, maši rukama dok objašnjavas, crtaj malo po ploči

16 slajd

Ovo je algoritam koji smo mi osmislili. Baziran je na PSO, ali smo mu dodali lokalnu pretragu motiviranu lokalnom pretragom iz ABC. Razlika je što naša lokalna pretraga prteražuje 'orbite' najbolje čestice te time pronalazimo najoptimalniji smijer za buduće kretanje same čestice i cijelog roja.

17 slajd

*pročitaj slajd, maši rukama dok objašnjavas, crtaj malo po ploči, skrenut pažnju na **

18 slajd

Naš kod je pisan u programskom jeziku c te je testiran na računalim sa 4 jezgrenima g4-bitnim ***GHz procesorima, 8Gb RAM memorije te operacijskim sustavom Linux Mint. Kao generator pseudo slučajnih brojeva samo koristili Mersenne Twister kojem smo kao 'sjeme' dali trenutno procesorsko vrijeme.
Pošto se procesorsko vrijeme mjeri u sekundama, a neki izračuni traju manje od toga, prije ponovnog pokretanja algoritma čekamo jedmo sekundu kako bi sigurno dobili drukčije pseudo slučajne brojeve i time osigurali da su nam reuzltati stvarno ovisni o slučajnim brojevima.

19 slajd

Testiranje smo izvršavali u 3 faze i treća faza je bila ona koja je dala konačne rezultate.
Kao što smo vidjeli, algoritmi koje smo odabrali sadrže jako veliki broj parametara, stoga su prve dvije faze testiranja bile nužne kako bi odradili neke fiksne parametre s kojima bi mogli provesti testiranje.
Prva faza testiranja nam je pokazala da su ovi *(pokzat na prvu točku* paramteri oni na kojima se vidi lijep prijelaz u vremenskoj složenosti.
Druga faza, i njene podfaze, testiranja su za rezultat dale rezultat da na sljedecim parametrima postižemo najbolje vrijednosti. ABC-u smo dali malu prednost s brojem čestica koje sudjeluju u lokalnim pretragama, tj. njihovim 'velikim' brojem. Kao što vidimo, radijus lokalne pretrage u ABC je varijabilan, tj. kada se pronađe bolje rješenje od trenutnog smanjujemo još više prostor te pretrage u nadi da ćemo pronaći još bolje.

crtat na ploču kako smo dali prednost.
Ukratko opisat što je koja varijabla koja je navedena

20 slajd

U testiranju smo koristili 10 testnih funkcija, a sada ćemo predstaviti smao nekoliko njih koje daju osnovnu predodžbu o tome na kakvim smo funkcijama testirali.
Prva i najjednostavnija funkcija je ova, sfera koja ima samo jedan lokalni, ujedno i globalni minimum.

21 slajd

Sljedeća funkcija je step funkcija. Problem koji se ovdje javlja je taj da imamo 'velike' dijelove di je funcija konstantna pa umjesto da napredujemo prema minimumu možemo zaglaviti na tim 'ravnim' djelovima.

22 slajd

Schafferova F6 funkcija. Problem predstavlja njezina 'gusta valovitost' i to što su lokalni ekstremi tih valova relativno blizu pa možemo u pretrazi lako preskočit optimum.

23 slajd

'Najgora' funkcija na kojoj smo testirali. Objedinjuje sve karakteristike koje imaju ostale funkcije nad kojima smo testirali.
Nacrtat presjek na ploču
Ima jako puno lokalnih minimuma ali optimum se nalazi na dnu jednog uskog lijevka koji je također jako 'nazubljen' od lokalnih minimuma.

24 slajd

Malo o dobivenim rezultatima. Pošto je bilo jako puno testiranja i rezultata prikazat ćemo samo rezultate dobivene na maloprije opisanim funkcijama. Ovo je pregled broja iteracija potrebnih za pronalazak optimuma na danu točonst (10^{-3} do 10^{-6}, ovisno o algoritmu). U svakoj ćeliji tablice su redom prikazani rezultati pri testiranju za $n = 50, 100, 150, 200$. Zelenom bojom smo označili najbolji rezultat koji smo dobili za svaki od testiranih broja čestica. Za ABC i Ackley funkciju nemamo podatke jer je izračun trajao predugo i nismo imali mogućnost to do kraja ispitati taj slučaj, tj. pristup računalima na kojima smo testirali je bio ograničen na rad i potrebu računala u praktikumu 5.

25 slajd

Ovdje vidimo grafički prikaz gore opisanih rezultata za Step funkciju i 100 čestica korištenih prilikom testiranja. Primjetimo kako je OPSO najstabilniji što se tiče broja potrebnih iteracija, dok je ABC jako ovisan o svojoj stohastičkoj naravi.

26 slajd

Evo i jedan grafički prikaz vremena izvršavanja algoritama nad Schafferovom F6 funkcijom i 50 čestica. Primjetimo da vrijeme izvršavanja izgleda zaista slučajno, a razlog tomu je stohastička narav algoritama, tj. prvi raspored čestica po prostoru pretraživanja.

27 slajd

ukratko o svakoj točki, ne ih čitat cijele to nema smisla

28 slajd

Grafičko sučelje koje smo osmisli i izradili bazirano je na web tehnologijama, Php i JavaScript.

29 slajd

Korisniku pruža mogućnost unosa većine paramtera koji su potrebi za izvršavanje algoritama; odabir funkcije nad kojom se testira, broj iteracija, broj ponavljanja i za algoritme specifične parametre.
Rezultati testiranja prikazuju se grafički u obliku grafova te kao textualni rezultati.