Elektronski fakultet Niš

Primena Particle Swarm optimizacije

Student: Miljana Randjelovic
Br. Indeksa: 1424

Sadržaj

1. Uvod

Modeliranje ponašanja prirode je jedno od najintuitivnijh nacina istrazivanja danasnjice. Mnogi algoritmi su implementirani tako da oponašaju prirodu i na taj nacin dodju do efikasnog resenja nekog problema. Jedna od najboljih tehnika danas za kreiranje algoritama zasnovanih na dešavanjima u prirodi se odnosi na Swarm inteligenciju. Swarm algoritmi podrazumevaju nekoliko pristupa u optimizaciji koji se mogu koristiti medjutim ni jedan od njih se ne smatra idealnim za svaki dati slucaj. U ovom dokumentu obradicemo konkretno primenu Particle Swarm optimiziacije (PSO) i njenu prirodu resavanja problema optimizacijom roja cestica. Ova optimizacija pripada oblasti racunarstva inspirisanog prirodom. Ova optimizacija ukratko trazi najpribliznije resenje u prostoru tako sto se iterativno pokusava poboljsati polozaj pojedinacne cestice.

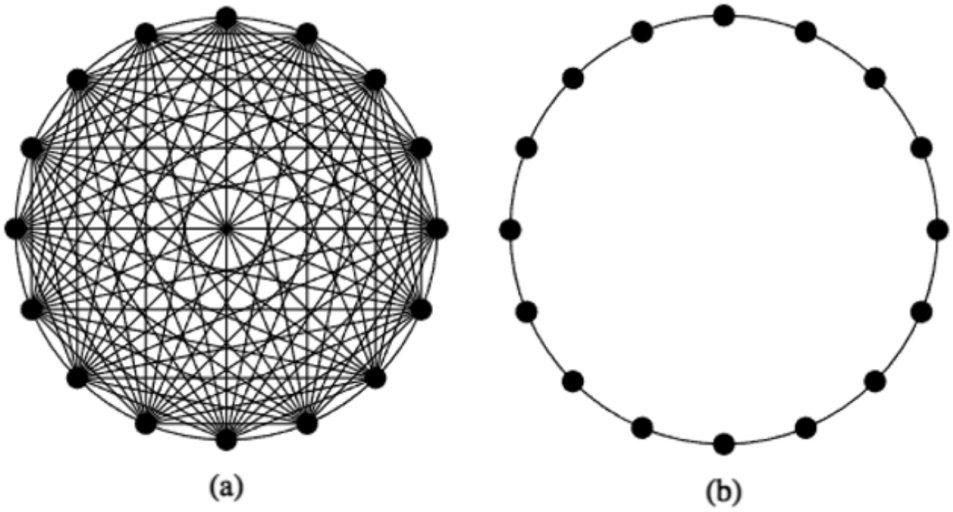
Na kraju cemo videti prakticnu implementaciju PSO algoritma koristeci python paket za PSO optimizaciju PySwarm.

2. Particle Swarm Optimizacija (PSO)

Nekoliko istrazivanja koja su sprovedena pocetkom 1990-ih godina a koja su se odnosila na istrazivanja opstanka zivotinja u prirodi i njihovog medjusobnog kretanja bila su okidac u nastanku algoritama koji ce za ideju implementacije koristiti upravo osobine koje poseduju grupe zivotinja poputa mrava ili jato ptica. Dakle kretanje pojedinacne jedinke u grupi u potrazi za hranom moze koristiti drugim jedinkama u odabiru pravca kretanja. Ovakva ideja iskoriscena je i za implementaciu PSO algoritama.

Particle swarm optimizacija zasniva se na konvergeniciji nasumicno odbaranih cestica ka lokalnom minimum. Konvergenicija cestica se desava kada se svi rekordi priblizavaju lokalnom minimum problema. Sposobnost PSO algoritma uveliko zavisi od topologije, dakle sa razlicitom strukturom brzina konvergencije algoritma i sposobnost da se izbegne preuranjena konvergencija na istom problemu optimizacije bice razlicite jer struktura topologiej odredjuje briznu deljenja informacija o pravcu kretanja izmedju svake cestice. Toplogija moze biti prstenasta i u takvom slucaju sve cestice su povezane medjusobno i dobija se najkraca prosecna udaljenost u roju, dok u topologiji tipa prstena svaka cestica je povezana sa najvise dva svoja suseda i dobija se najveca prosecna udaljenost u roju.

Dve najcesce koriscene topologije u PSO optimizaciju jesu globalna struktura zvezde i lokalna prstenasta struktura. U lokalnoj strukturi najbliza cestica je odredjena indeksom. U nastavku su prikazane dve najcesce koriscene arhitekture:



Slika 1: Topologija zvezde i prstena

3. Kako radi Particle Swarm optimizacija ?

Particle Swarm optimizacija se prvi put pominje 1995 godine u radu Kennedy & Eberhart. PSO ima mnogo prednosti, jednostavan je, brz i moze se kodirati u nekoliko redova. Takodje, zahtevi za memorijom su minimalni u okviru PSO algoritma. Ovaj algoritam ima prednosti u odnosu na evolucione i genetske algoritme. Najpre, PSO poseduje memoriju, odnosno svaka cestica pamti svoje najbolje resenje i to licno najbolje *pBest* kao i grupno najbolje resenje *gBest.* Jos jedna prednost je da se odrzava pocetna populacija pa nema potrebe za primenom operatora na populaciju sto je proces koji oduzima dosta vremena i memoriju.

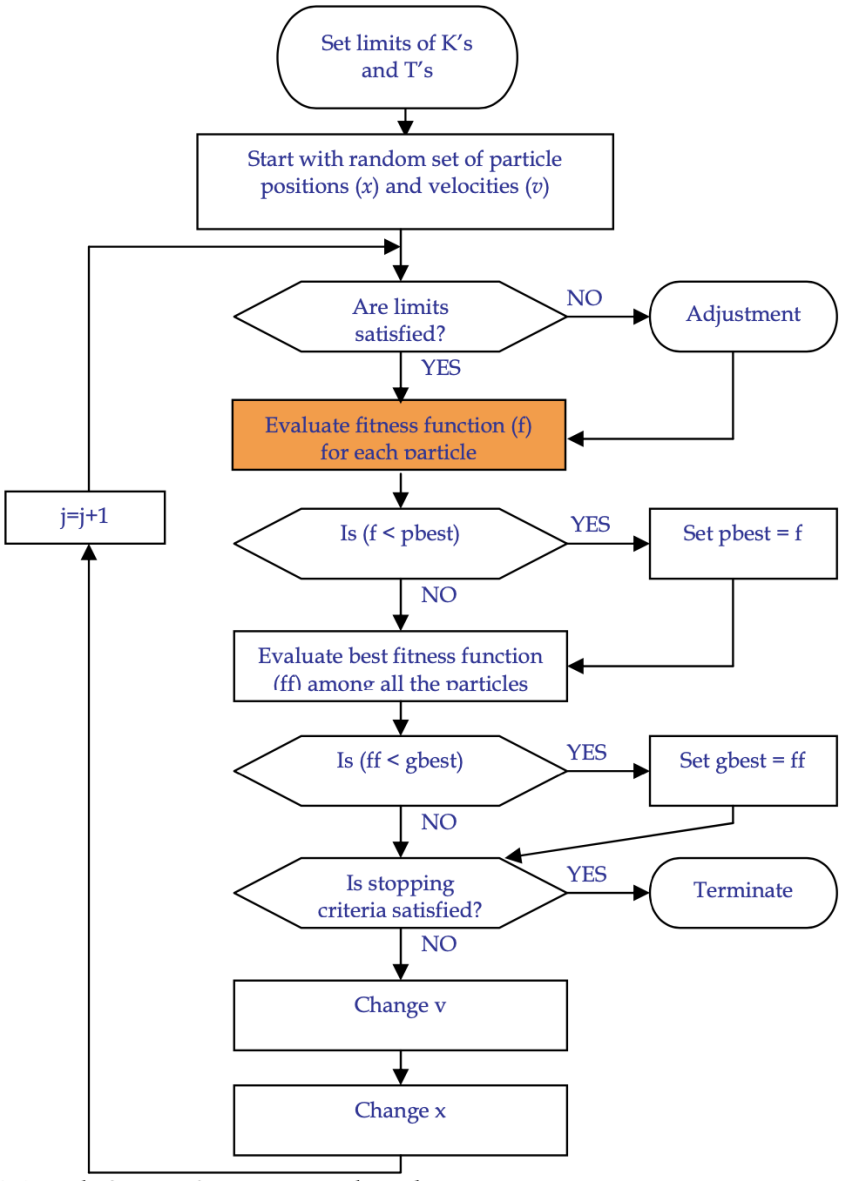
PSO pocinje sa populacijom nasumicnih cestica resenja u prostoju *D* dimenzija. Svaka I_{th} -a cestica predstavljena je sa *Xi=(xi1, xi2, ..., xiD)*. PSO algoritam u svakom koraku menja brzinu svake cestice (*velocity*) ka njenom *pBest* i *gBest* resenju prema jednacini (21). Brzina cestice *i* je predstavljena kao *Vi=(vi1,vi2, …, viD).* Pozicija svake cestice se azurira po jednacini (22).

$$v_{id} = wv_{id} + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (p_{gd} - x_{gd}) \quad (21)$$

$$x_{id} = x_{id} + v_{id} \quad (22)$$

$p_{id} = pbest$ and $p_{gd} = gbest$

U nastavku je prikazan PSO algoritam:



Korak 1: Definisati prostor problema i postaviti prihvatljive granice parametara
Korak 2: Inicijalizovati niz cestica sa nasumicno dodeljenim pozicijama i brzinama u prostoru problema. Ove cestite predstavljaju inicijalni skup resenja.
Korak 3: Proveriti da li je trenutna pozicija cestice u prostoru problema. Ukoliko nije podesiti novu poziciju kako bi cestica bila u prostoru problema.
Korak 4: Proceniti *fitness* vrednost svake cestice.

Korak 5: Porediti trenutnu vrednost *fitness-a* sa starom vrednoscu (*pBest*). Ukoliko je trenutna vrednost bolja onda uzeti novu vrednost kao trenutni *pBest* i postaviti trenutne koordinate na *pBestXi.*
Korak 6: Odrediti trenutni globalni minimum medju najboljom pozicijom cestice.
Korak 7: Ako je trenutni globalni minimum bolji od *gBest* onda postavi novu vrednost *gBest* na trenutni globalni minimum i postaviti koordinate trenutnog globalnog minimuma na *gBestXi.*
Korak 8: Promeniti brzinu (*velocity*) po jednacini (21)
Korak 9: Pomeriti svaku cesticu na novu poziciju dobijenu iz jednacine (22)
Korak 10: Ponoviti korake od 3 - 9 sve dok se ne zadovolje kritetrijumi za kraj.

Kako bi se postigla istovremena stabilizacija, razmatra se istovremeno nekoliko radnih tacaka. Za svaku radnu tacku se izracunava cilj *Ji.* Onda *J = max(J1, J2, …, Jnop)* gde je *Nop* broj radnih tacaka koji se ocenjuje.

4. Razlicite varijante Particle Swarm optimzaije

Postoji nekoliko varijanti Particle Swarm optimizacije, neke od njih podrazumevaju razlicit nacin izracuvanja startne brzine i odabir startnih pozicija. Neke od varijanti su Hibridni (Hybrid PSO) i Postepeni (Gradient PSO).

Hibridni PSO: U cilju povećanja performansi optimizacije uvode se nove i naprednije PSO varijacije. Postoje određeni pomaci u toj studiji kao što je razvoj hibridnog pristupa optimizacije koji kombinuje PSO sa drugim optimizatorima. Jedan od njih je kombinovanje PSO sa optimizacijom zasnovanom na biogeografiji i uključujući efikasan mehanizam učenja.

Gradient PSO: Da bi se konstruisali PSO algoritmi zasnovani na gradijentu, sposobnost PSO algoritma da efikasno istražuje mnoge lokalne minimume može se kombinovati sa sposobnošću algoritama lokalnog pretraživanja zasnovanog na gradijentu da efikasno izračunaju tačan lokalni minimum.

5. Implementacija PSO uz pomoc PySwarms

PySwarms je alatka zasnovana na Python-u za optimizaciju roja čestica. PySwarms nudi interakciju sa optimizacijama roja i osnovnu optimizaciju sa PSO. PySwarms implementira tehnike optimizacije roja sa više čestica na visokom nivou. Kao rezultat

toga, teži da bude lak za upotrebu i prilagodljiv. Pomoćni moduli se takođe mogu koristiti da vam pomognu sa konkretnim problemom optimizacije. Primer implementacija PSO optimizacije dodat je na git repozitorijum.

6. Literatura

1. ACO Based Shortest Path between Locations within a Campus, Maria Wenisch S 1 , Amali Asha A 2., Loyola Institute of Business Administration, Chennai, India 2 Loyola College, Chennai, India
2. A Tutorial on Particle Swarm Optimization: https://analyticsindiamag.com/a-tutorial-on-particle-swarm-optimization-in-python/
3. A Gentle Introduction to Particle Swarm Optimization: :https://machinelearningmastery.com/a-gentle-introduction-to-particle-swarm-optimization/