

Određivanje putanja naelektrisane čestice u električnom polju primenom genetičkog algoritma (2D slučaj)

Analizirana je primena genetičkog algoritma u određivanju optimalnih putanja naelektrisane čestice u nehomogenom električnom polju. Dobijena je zavisnost srednjeg dejstva (funkcije prilagođenosti) od broja generacija, procentni učinak svake od mogućih putanja i minimalno dejstvo svake od njih. Zaključeno je da dovoljnim brojem ponavljanja genetičkog algoritma možemo sa zadovoljavajućom tačnošću naći većinu lokalno optimalnih putanja i da, iako ne možemo biti sigurni da smo ih našli sve, možemo sa sigurnošću naći najverovatnije.

Uvod

Optimizacija predstavlja određivanje vrednosti jednog ili više parametara, u cilju nalaženja minimuma ili maksimuma date funkcije. Jedna od metoda optimizacije je genetički algoritam koji imitira ponašanje DNK u prirodi. Njegov zadatak je da optimizuje funkciju prilagođenosti $f(x)$, gde je parametar x string (naš "hromozom"). Cilj nam je da nađemo bitovni string tj. hromozom koji ima najbolju prilagođenost. Prilagođenost (*fitness*) nalazi se primenjivanjem funkcije prilagođenosti na bitovni string (MacKeown 1997).

Prednost genetičkog algoritma je to što nije bitno da unapred znamo kako se ponaša funkcija koju treba optimizovati. Sa zadovoljavajućom tačnošću on nam može naći rešenja veoma složenih problema.

Mane genetičkog algoritma su to što nikad ne dobijamo apsolutni minimum funkcije, već rezultat osciluje u okolini minimuma. Takođe, za neke jed-

nostavnije probleme, postoje mnogo efikasnije metode za nalaženje rešenja – genetički algoritmi su spori.

Za implementaciju genetičkog algoritma potrebno je sledeće:

1. specifikacija parametara tako da liče na hromozome
2. recept za mutacije i/ili reprodukciju hromozoma
3. intervencija koja modelira prirodnu selekciju i bira hromozome sa boljom vrednosti funkcije prilagođenosti za dalju reprodukciju.

Genetički algoritmi mogu biti veoma složeni. Ako nismo dovoljno pažljivi, može da se desi da program upadne u beskonačnu petlju i nikad ne završi započeti posao.

Ovako izgleda genetički algoritam dat korak po korak (Reitman 1994):

1. Inicijalizacija populacije hromozoma.
2. Procena svakog hromozoma populacije.
3. Stvaranje novih hromozoma korišćenjem onih koji su trenutno na raspolaganju. Kod reprodukcije se primenjuju mutacije i rekombinacije (*crossing-over*).
4. Brisanje manje prilagođenih članova populacije kako bi se oslobodilo mesto za nove hromozome.
5. Procena novih hromozoma i njihovo ubacivanje u populaciju.
6. Ako je kriterijum zaustavljanja zadovoljen, onda stajemo. Ako nije, povratak na korak 3.

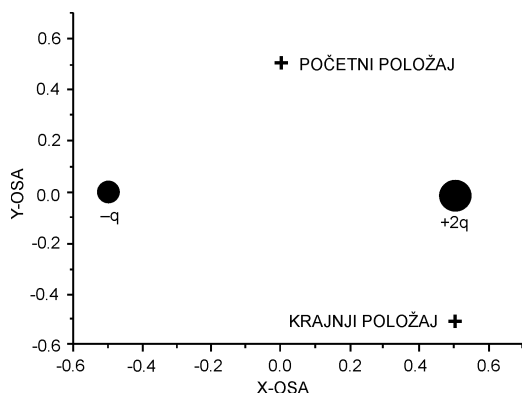
Sistem koji smo posmatrali nalazi se u ravni (slika 1). Dva naelektrisanja, $(+q)$ i $(+2q)$ imaju fiksni položaj. Treće naelektrisanje $(-q)$ ima određeni početni i krajnji položaj, ali njegova putanja od jednog do drugog može da se menja. Zadatak je da se pronađu sve moguće putanje od početne do krajnje tačke. Ovo je rađeno genetičkim algoritmom.

Zbog postojanja naelektrisanih čestica, svaka tačka u ravni ima električni potencijal (slika 2). čestica kreće iz početnog položaja i teži da se kreće putanjom što manje promenljivog potencijala. To nas navodi da očekujemo 3 klase mogućih putanja: "levo", "kroz sredinu" i "desno".

Marija Vranić (1984), Beograd, Ljutice Bogdana 2/3/71, učenica 4. razreda Matematičke gimnazije u Beogradu

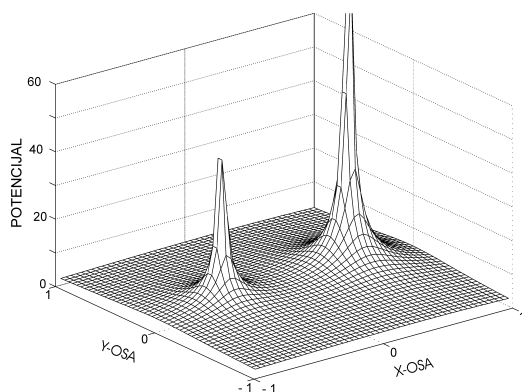
MENTOR:

Mr Srdan Verbić, Istraživačka stanica Petnica



Slika 1.
Skica sistema

Figure 1.
Model of the system



Slika 2.
Mapa potencijala za sistem dat na slici 1

Figure 2.
Potential map of the system shown in Figure 1

Opis metode

Specifikacija parametara. Moguće putanje predstavljaju hromozome za genetički algoritam. One su date u obliku niza tačaka. Koordinate jedne tačke predstavljaju gen.

Generisanje početne konfiguracije. Putanja se generiše tako što se između dve tačke ubacuje treća. Tačka se ubacuje na simetralu duži tako da njena udaljenost od preseka simetrale i duži nije veća od dvostruke dužine duži. Udaljenost od preseka dobija

se množenjem maksimalne udaljenosti sa slučajnim brojem iz intervala $[0,1]$.

Procena prilagođenosti. Funkcija prilagođenosti u ovom slučaju je dejstvo. Putanje su utoliko bolje ukoliko imaju manje dejstvo.

$$\int_p^k (T - U) ds = \min$$

Ovde je p početni, k krajnji položaj, T kinetička, U potencijalna energija, a ds rastojanje između dve susedne tačke. Izračuna se dejstvo svake putanje i one se sortiraju po tome koliko su dobre. Računa se i prosečno dejstvo za datu populaciju. Po završenoj evaluaciji, odbacuje se polovina populacije koja ima lošiju prilagođenost.

Crossing-over. *Crossing-over* je razmena gena dva hromozoma. Oni se biraju slučajno, kao i gen na kome će se izvršiti razmena. To izgleda ovako:

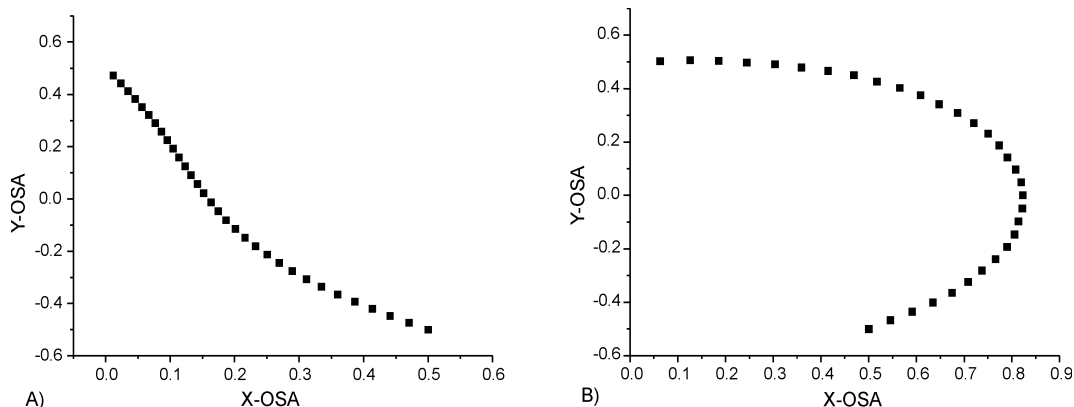
$$(110'00) * (011'01) \rightarrow (11001) + (01100)$$

Jedina razlika je u tome što su kod nas geni koordinate. Novonastali hromozomi dolaze na mesto odbačenih.

Mutacije. Mutacija predstavlja promenu samo jednog gena u okviru jednog hromozoma bez ikakvog uticaja drugih jedinki populacije. Realizuje se slično generisanju, samo što ne nastaje nova tačka, već se postojeća tačka pomera tako da bude na simetrali duži koju čine njoj susedne tačke. Razlika se ogleda i u tome što je maksimalna udaljenost tačke od preseka simetrale i duži jednaka veličini mutacije, a ne dvostrukoj dužini duži.

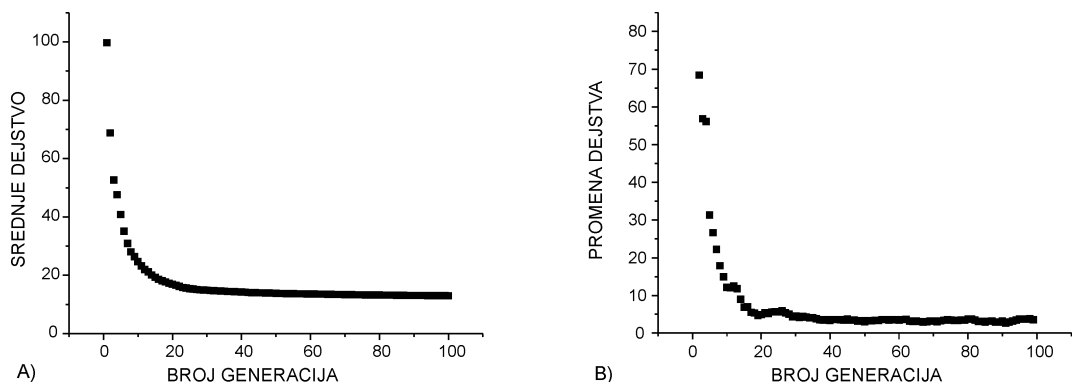
Kriterijum zaustavljanja. Za ovo postoje dve mogućnosti. Program se može pustiti da radi dok se varijacija dejstva dovoljno ne smanji, ali pri tome se mora voditi računa o veličini mutacija i tome da one ne prave veće varijacije dejstva od kritične vrednosti. Takva greška bi mogla dovesti do beskonačnog ciklusa. Druga varijanta je da se unapred odredi broj generacija koje će program da simulira, što ima smisla jer posle određenog broja generacija dejstvo konvergira. Ovde se mora izabrati dovoljno veliki broj generacija.

Genetički algoritam nalazi jedan lokalni minimum *fitness* funkcije. Ponavljanjem algoritma nalazimo više lokalnih minimuma. To ne znači da će algoritam svaki put naći novi lokalni minimum. Naprotiv, mnogo puta će se ponoviti vrednost koju smo već dobili (tj. vrednost bliska nekoj od dobi-



Slika 3. Optimalne trajektorije “kroz sredinu” (A) i “desno” (B)

Figure 3. Optimal paths “in between” (A) and “right” (B)



Slika 4. Zavisnost srednjeg dejstva (A) i promene dejstva (B) od broja generacija

Figure 4. Average value (A) and variations (B) of least action integral as a function of number of generations

jenih), ali to je samo još jedna potvrda da smo našli lokalni minimum. Ukoliko se neki minimum više puta ponavlja, onda je on verovatniji. Primitimo da ovom metodom ne možemo biti sigurni da smo našli sve lokalne minimume, ali možemo sa sigurnošću naći najverovatnije od njih.

Za implementaciju ovog algoritma potrebna je random funkcija koja generiše slučajne brojeve iz intervala $[0,1]$ po uniformnoj raspodeli.

Rezultati i diskusija

Program daje sve tri klase putanja, ali ne sa istom verovatnoćom. Putanja “kroz sredinu” se po-

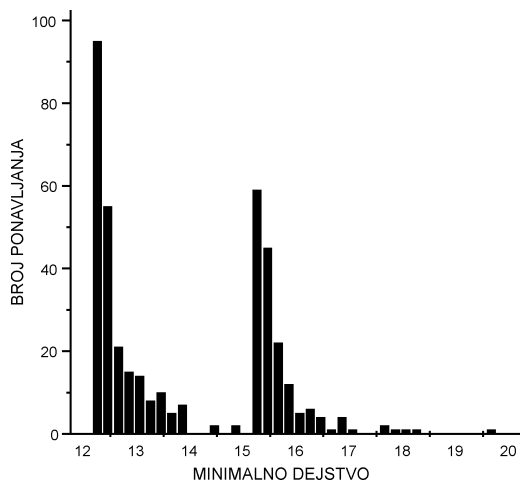
javljuje najčešće, zatim putanja “desno”, dok se putanja “levo” pojavljuje jako retko.

Minimalno dejstvo ovih putanja je:

putanja	dejstvo
“kroz sredinu”	12.4 ± 0.3
“desnu”	15.5 ± 0.3
“levo”	18.2 ± 0.8

Kao što je očekivano, srednje dejstvo konvergira posle određenog broja generacija (slika 4a).

Posmatrana je i promena dejstva u zavisnosti od broja generacija (slika 4b). Ono opada do neke vrednosti (različite od nule) i posle osciluje oko nje, zavisno od veličine mutacija. Naime, posle nekog



Slika 5. Histogram minimalnog dejstva u 400 iteracija

Figure 5. Histogram of least action integral for 400 iterations

vremena sve putanje budu jako slične, ali mutacije ih u svakom koraku menjaju tako da diverzitet ne može da padne do nule. Izborom manjih mutacija možemo njegovu asimptotu da približimo nuli.

Na histogramu postignutog minimalnog dejstva u velikom broju testova (slika 5) jasno su uočljive dve vrednosti oko kojih se grupišu tačke. Najmanje dejstvo odgovara putanji “kroz sredinu”, a sledećec putanji “desno”. Dejstvo koje odgovara putanji “levo” dobija se u jako malom broju slučajaja (3%). To znači da je verovatnoća da naelektrisana čestica prođe tim putem jako mala. Putanja “kroz sredinu” pojavljuje se u 60% slučajaja, a putanja “desno” u 37% slučajaja.

Greške. Konačna putanja koja se dobija nema apsolutni minimum dejstva jer mutacije stalno imaju istu maksimalnu vrednost, pa i putanja ima bar toliku grešku. Grešku je određena kao standardna devijacija dejstva za određenu klasu putanja.

Zaključak

Genetički algoritam je pouzdana metoda optimizacije. Skoro uvek se može postići željena tačnost u nalaženju ekstremuma, što nam je dovoljno za praktičnu primenu. Ponavljanjem postupka mogu se naći i svi ili skoro svi lokalni ekstremumi. Problem kretanja čestice u nehomogenom električnom polju može da se proširi na opštiji slučaj sa naelektrisanim česticama u ravni ili prostoru. Takođe je moguće razmatrati kretanje u promenljivom električnom polju ako je promena polja predvidiva.

Literatura

- MacKeown K. 1997. *Stochastic Simulation in Physics*. Springer
- Reitman E. 1994. *Genesis Redux: Experiments Creating Artificial Life*. McGraw-Hill

Marija Vranić

Determination of Charged Particle's Trajectory in Electrical Field using Genetic Algorithm (2D Case)

We test the efficiency of the Genetic Algorithm (GA) optimization in finding the path between two fixed spots which has the minimal least action integral was tested. The system consisted of two immobile point charges situated in a plane, and a third point charge which had to find the best way from initial to the final location. GA method was used for the determination of all local minimums. Theoretically expected paths were found. This GA method has been concluded to be reliable for finding equilibrium states. In this case, three equilibrium states and their least action integral were found. This problem can be generalized by adding more charges on a plane and/or considering it in a multi-dimensional space. This method could be used for research of moving systems, provided that we have enough information about the system.

