Univerzitet u Novom Sadu

Fakultet tehničkih nauka

Dokumentacija za projektni zadatakkk

Studenti: Mila Milović, SV22/2021

Sonja Baljicki, SV59/2021

Predmet: Nelinearno programiranje i evolutivni algoritmi

Broj projektnog zadatka: 8

Tema projektnog zadatka: Genetski algoritam, Fišerov nasumični šah

Opis problema

Potrebno je rešiti problem Fišerovog nasumičnog šaha(Šah960), koji je izmislio američki šahista, velemajstor i bivši svetski šampion u šahu Bobi Fišer. Ovaj problem se naziva Šah960 zato što postoji 960 mogučih kombinacija koje igrači mogu da dobiju.

Ograničenja za rešavanje problema su sledeći:

* Pešaci se nalaze na uobičajenim poljima
* Ostale figure su u prvom redu
* Kralj se postavlja na bilo kom polju između dva topa, ne sme da se postavi na polja a i h jer onda ne bi bilo mesta za topa
* Lovci se postavljaju na dva polja različite boje

Treba da se ispoštuju sva zadata ograničenja i da se nađe zadovoljavajući raspored figura. Kriterijum optimalnosti je da raspored figura bude što bliži rasporedu figura u redovnom šahu. Zbog toga što raspored figura u klasičnoj verziji šaha zadovoljava ograničenja problema, algoritam najčešće i nađe taj raspored. Za implementaciju će se koristiti genetski algoritam.

Uvod

Genetski algoritam je algoritam inspirisan teorijom prirodne evolucije Čarlsa Darvina. Ovaj algoritam oponaša proces prirodne selekcije gde se najsposobniji pojedinci biraju za reprodukciju, ukrštanje, kako bi proizveli potomstvo sledeće generacije, tako da im fitness vrednost(mera njihove sposobnosti) bude što bolja. Postupak genetskog algoritma:

* Inicijalizacija početne populacije
* Računanje fitnes funkcije
* Iterirati:
  + Selekcija
  + Ukrštanje
  + Mutacija
  + Računanje fitnes funkcije

Implementacija

* Implementacija započinje definisanjem jedinki koje su potrebne za dalji rad genetskog algoritma. Jedna jedinka predstavlja niz brojeva od 1 do 5, gde je 1:top, 2:konj, 3:lovac, 4:kraljica i 5:kralj. Prva generacija se formira tako što se napravi niz od 10 elemenata koji su permutacije osnovnog rasporeda figura.
* Permutacije implementiramo uz pomoć funkcije, koja je napravljena specifično za ovaj problem, da ne bi dolazilo do velikog broja nevalidnih rasporeda, što bi dalje izazvalo veoma dugo izvršavanje programa. Funkcija se sastoji iz nekoliko celina:
  + Prvo se postavlja kralj da nasumičnu poziciju sa indeksom između 1 i 6, što je bilo koja pozicija osim krajnjih.
  + Nakon toga postavljaju se topovi, jedan po jedan, tako što se prvi postavi na nasumičnu poziciju levo od kralja, a drugi desno.
  + Lovci se postavljaju na nasumične pozicije, sve dok indeksi ne budu različito kongruentini po modulu 2, odnosno jedan lovac je na crnom, a drugi na belom polju.
  + Nakon postavljalja lovaca, postavlja se kraljica na nasumičnu slobodnu lokaciju.
  + Na kraju postavljamo konje na preostale slobodne pozicije.
* Posle generisanja početne populacije ulazimo u petlju genetskog algoritma, gde se prvo rangira populacija. Funkcija rangiranja ima povratnu vrednost listu koja sadrži jedinke iz početne populacije, soritrane u rastućem redosledu po kriterijumu fitnes vrednosti.
* Fitnes vrednost se računa kao zbir razdaljina pozicije svake figure od njene pozicije u klasičnom rasporedu šaha, zbog toga što je kriterijum optimalnosti što sličniji raspored redovnom šahu.
* Rulet selekcija omogućava da se od populacije formiraju parovi roditelja, koji će se u narednoj fazi ukrštati da bi se dobila sledeća generacija. U rulet selekciji se fitnes vrednost svake jedinke množi sa nasumično odabranom vrednošću, što dovodi do raznovrsnosti parova. Ovim se izbegava situacija gde genetski algoritam bira samo najbolje jedinke. Takođe, zajedno sa mutacijom sprečava da se algoritam završi na lokalnom umesto na globalnom optimumu.
* Sledeća faza je ukrštanje prethodno napravljenih parova roditelja, tako što se od jednog para roditelja dobije jedno dete. Prvi pokušaj implementacije je funkcionisao na sledeći način: poslednjih n pozicija jednog roditelja se menjalo sa poslednjih n pozicija drugog, pri čemu su formirana dva deteta. Ipak na ovaj način su se dobijala deca sa rasporedom figura koji nije odgovarao uslovima Fišerovog šaha, ali još gore deca nisu imala odgovarajući broj svake od figura. Nije pomoglo ni stavljanje ukrštanja u for petlju koja se završavala tek kad se pronađe validno rešenje, zbog toga što se ova for pelja izvršavala nekoliko minuta. Iz gore navedenih razloga, sastavljena je posebna funkcija za ukrštanje. Njen algoritam je sledeći:
  + Prvo se proveri da li je kralj prvog roditelja na poziciji koja je između pozicija topova drugog roditelja. Ako jeste, ovako se postavljaju kralj i topovi. Ako nije proverava se suprotno. Ako ni ovo nije validno i kralj i topovi se postave na pozicije kao kod prvog roditelja.
  + Proverava se da li se na tablu mogu postaviti lovci, jedan od jednog, a drugi od drugog roditelja. Ako to nije moguće pokušava se postavljanje lovaca na pozicije drugog roditelja, zatim na pozicije prvog. Ako ni ovo nije moguće lovci se postave na nasumične pozicije koje zadovoljavaju uslov da se nalaze na poljima različite boje.
  + Nakon postavljalja lovaca, pokušava se postavljanje kraljice na poziciju drugog roditelja, pa onda na poziciju kraljice kod prvog roditelja. Ukoliko nijedan od prethodnih načina nije uspešan, kraljica se postavlja na nasumičnu poziciju,
  + Na kraju postavljamo konje na preostale slobodne pozicije.
* Posle ukrštanja sledi mutiranje dobijene dece. Mutacija nam omogućava genetsku raznovrsnost. Ona se odvija tako što za svako dete postoji mala verovatnoća da dve figure zamene mesta. Verovatnoća je postavljena na 0.1 zbog toga što bi veća verovatnoća proizvela suviše nevalidnih pozicija tj. rasporeda, a manja bi dovela do toga da mutiranje nema dovoljno veliki uticaj.
* Elitizam tj formiranje nove generacije omogućava da se od roditelja i dobijene dece sastavi nova početna popupacija za sledeću iteraciju. Parametar funkcije koja vrši elitizam je koeficijent koji predstavlja koji procenat dece, a koji procenat roditelja sačinjava novu generaciju. Ispitivanjem programa zaključeno je da koeficijent 0.5 najbolje odgovara problemu Fišerovog šaha.

Zaključak

Program poseduje nekoliko parmetara, čije su vrednosti odabrane putem testiranja:

* Veličina populacije 🡪 10
* Broj iteracija glavne for petlje 🡪 50
* Koeficijent mutacije 🡪 0.1
* Koeficijent elitizma 🡪 0.5

Nakon izvršavanja programa crta se grafik gde su na x osi brojevi iteracija, a na y osi fitnes vrednosti jedinki. Ovaj grafik nam pomaže u analizi efiksnosti programa u pronalaženju rešenja, jer je jednostavno pročitati ka kojoj vrednosti jedinke konvergiraju.

Rezultat izvršavanja programa je najčešće jedinka sa fitnes vrednošću u intervalu [0,4], zbog toga što su to ti redosledi koji su najbliže redovnom redosledu, a to i jeste kriterijum optimalnosti.

# Neki od rezultata rada programa su:

♖ ♘ ♔ ♕ ♗ ♗ ♘ ♖ --> fitnes vrednost je 4

♖ ♘ ♗ ♔ ♕ ♗ ♖ ♘ --> fitnes vrednost je 4

♖ ♘ ♗ ♔ ♕ ♗ ♘ ♖ --> fitnes vrednost je 2

♖ ♘ ♗ ♕ ♔ ♗ ♖ ♘ --> fitnes vrednost je 2

♖ ♘ ♗ ♕ ♔ ♗ ♘ ♖ --> fitnes vrednost je 0

Sa obzirom na podatak da je rezultat programa u otprilike 50% slučajeva raspored sa fitnes vrednošću 0 možemo zaključiti da je rešavanje ovog problema genetskim algoritmom daje zadovoljavajuće rezultate.