

5. domaća zadaća – inačice genetskog algoritma

Zadatak 1.

Napisati genetski algoritam koji se temelji na RAPGA pristupu i koji rješava modificirani problem *Max-Ones*. Taj problem možemo opisati na sljedeći način. Radimo s vektorom bitova (označimo ga v) duljine n . Označimo s k broj bitova u vektoru v koji su postavljeni na 1. Funkcija $f(v)$ definirana je po dijelovima kako slijedi.

- Za k manji ili jednak $0.8 \cdot n$ vrijedi $f(v) = k / n$.
- Za k između $0.8 \cdot n$ i $0.9 \cdot n$ vrijedi $f(v) = 0.8$.
- Za k veći od $0.9 \cdot n$ vrijedi $f(v) = (2k/n) - 1$ (tj. linearno nastavlja rasti od 0.8 do 1 kada se k izjednači s n).

Zadatak je pronaći vektor v koji maksimizira funkciju $f(v)$. Iako je nama rješenje ovog problema evidentno, genetskom algoritmu baš i neće biti, posebice zbog platoa koji funkcija poprima u jednom dijelu. Skicirate li graf funkcije dobrote u ovisnosti o broju jedinica koje su postavljene u rješenju, uočite ćete da se ona u jednom dijelu ne mijenja: broj jedinica se povećava (što je pravi smjer prema optimalnom rješenju) no funkcija dobrote to ne oslikava. Ovakvi problemi mogu biti dosta teški za rješavanje jer jednom kada GA rješenja dovede do tog platoa, funkcija dobrote mu više ni na koji način ne pomaže (sve do izlaska iz platoa). Prilikom rješavanja ovog problema nije dozvoljeno u algoritam ugrađivati "heurističke" metode -- pustite sam algoritam da pokuša pronaći rješenje. Program mora primiti n kao parametar. Isprobajte kako se program ponaša za $n=10$, $n=100$ te $n=1000$.

Podsjetnik na RAPGA: algoritam kao parametre treba ogradu za minimalnu populaciju, ogradu za maksimalnu populaciju, maksimalni selekcijski pritisak te mehanizam koji određuje kako će se mijenjati (i hoće li) *CompFactor* temeljem kojeg se određuju uspješna djeca. Obratite pažnju da algoritam u populaciju djece ne smije stavljati duplikate: u okviru ove zadaće radite provjeru na razini genotipa, i pokušajte to napraviti efikasno.

Implementirajte dvije vrste selekcije (i isprobajte rad te usporedite rezultate sa svakom od njih):

1. Za selekciju oba roditelja radite k -turnirsku selekciju (ovaj k nema veze s onim iz funkcije dobrote).
2. Za selekciju jednog roditelja radite k -turnirsku selekciju a drugog roditelja odaberite potpuno slučajno.

Slučajeve 1 i 2 isprobajte za k : 2, 3 te 5. Ima li razlike?

Rješenje smjestite u razred `hr.fer.zemris.optjava.dz5.part1.GeneticAlgorithm`. Jedini ulaz algoritma preko komandne linije treba biti n ; sve ostalo postavite u kodu na vrijednosti koje smatrate primjerenima. U projekt dodajte datoteku `README.txt` i u njoj dajte odgovore na prethodna pitanja te pojasnite eventualne odluke koje ste donijeli vezano uz implementaciju algoritma a koje su ovom uputom ostale nedefinirane.

Zadatak 2.

Riješite zadatak *A.9* iz knjige (u vašem projektu napravite poddirektorij `data` i u njega spremite datoteke s primjerima koje trebate isprobati). Za potrebe rješavanja zadatka napišite *SASEGASA*. Rješenje smjestite u razred `hr.fer.zemris.optjava.dz5.part2.GeneticAlgorithm`. Pri pokretanju, program treba dobiti putanju do datoteke s problemom koji se rješava, ukupni broj jedinki te početni broj podpopulacija. Ako želite dodati još neki parametar, dokumentirajte ga. Algoritam treba na zaslon ispisati konačno rješenje i njegovu kaznu (uočite da je problem minimizacijski i da je propisan način kako računati kaznu).

Napomene:

Rok za predaju Eclipse projekta je sljedeći četvrtak do termina predavanja (14:00).