

## 9. domaća zadaća – višekriterijska optimizacija uporabom algoritma NSGA

### Uvod

U ovoj zadaći implementirat ćete algoritam NSGA koji obavlja višekriterijsku optimizaciju funkcije. Algoritam su 1994. predložili Srinivas i Deb.

U okviru ove domaće zadaće rješavat ćete optimizacijski problem prikazan u nastavku.

$$\text{Problem: } \begin{cases} \text{minimiziraj} & f_1(\vec{x}) = x_1, \\ \text{minimiziraj} & f_2(\vec{x}) = \frac{1+x_2}{x_1} \\ \text{uz ograničenje} & 0.1 \leq x_1 \leq 1, \\ \text{uz ograničenje} & 0 \leq x_2 \leq 5 \end{cases}$$

I prije samog postupka rješavanja, s obzirom na jednostavnost kriterijskih funkcija, možemo nešto zaključiti o njihovom odnosu. Vrijedi:

$$\begin{aligned} f_1 &= x_1 \\ f_2 &= \frac{1+x_2}{f_1} \end{aligned}$$

Stoga je lako zaključiti da će se uz fiksirani  $f_1$  minimum od  $f_2$  postići postavljanjem  $x_2$  na vrijednost 0 što je njegova donja granica. Tada će vrijediti:

$$f_2 = \frac{1}{f_1}$$

što je izraz iz kojega možemo zaključiti kakav oblik Pareto fronte očekujemo.

### Zadatak.

Napisati općenitu implementaciju algoritma NSGA. Pod pojmom općenitu podrazumijevamo implementaciju koja će moći raditi s proizvoljnim brojem kriterijskih funkcija. Prijedlog je problem višekriterijske optimizacije modelirati zasebnim sučeljem koje nudi barem dvije metode ("barem" jer će nedostajati još neke informacije o problemu koje možete po potrebi dodati):

```
interface MOOPProblem {
    int getNumberOfObjectives();
    void evaluateSolution(double[] solution, double[] objectives);
}
```

Metoda `getNumberOfObjectives()` vraća broj kriterija koji su definirani u okviru problema. Metoda `evaluateSolution` je zadužena za vrednovanje rješenja: prima točku iz prostora rješenja, nad njom računa vrijednosti svih kriterija i rezultat pohranjuje u polje koje je predano kao drugi argument. Umjesto predloženog oblika metode `evaluateSolution` možete koristiti i metodu oblika:

```
double[] evaluateSolution(double[] solution);
```

Jedina razlika je što prvi oblik dozvoljava da se polje u koje će biti pohranjen rezultat unaprijed stvori izvana i potom koristi svaki puta dok drugi oblik traži metodu `evaluateSolution` da svaki puta alocira novo polje u koje će pohraniti rezultat vrednovanja.

Također, pretpostavite da se po svim kriterijima **uvijek radi minimizacija**, odnosno da će, ako je problem drugačiji, on najprije biti pretvoren u minimizacijski problem.

Rad programa isprobajte na dva problema.

### Problem 1.

Ovaj problem definiran je na sljedeći način. Potrebno je minimizirati:

$$\text{minimiziraj } f_1(\vec{x}) = x_1^2$$

$$\text{minimiziraj } f_2(\vec{x}) = x_2^2$$

$$\text{minimiziraj } f_3(\vec{x}) = x_3^2$$

$$\text{minimiziraj } f_4(\vec{x}) = x_4^2$$

pri čemu je  $\vec{x} = (x_1, x_2, x_3, x_4) \in R^4$  uz ogradu:

$$-5 \leq x_i \leq 5, \quad i \in \{1, 2, 3, 4\}.$$

Razmislite li, brzo ćete uvidjeti da ovo nije uistinu višekriterijski problem (razumijete li zašto?). Poslužit će nam, međutim, za provjeru rada implementacije.

### Problem 2.

Fokus ove domaće zadaće je primjena na problem opisan u uvodnom dijelu. Taj problem jest uistinu višekriterijski što smo u uvodnom dijelu i eksplicitno pokazali (gdje odnosno kako?).

### Rješenje

Napišite program `hr.fer.zemris.optjava.dz9.MOOP` koji preko komandne linije prima sljedeće argumente:

- *fja*: 1 za prvi problem, 2 za drugi problem,
- *n*: željena veličina populacije,
- *vrsta*: prostor u kojem se računa funkcija dijeljenja,
- *maxiter*: maksimalni broj generacija/epoha.

Kao drugi argument (*vrsta*) programa potrebno je prihvatiti sljedeće:

- decision-space: funkciju dijeljenja dobrote potrebno je računati u prostoru rješenja;
- objective-space: funkciju dijeljenja dobrote potrebno je računati u prostoru kriterijskih funkcija.

Po završetku rada optimizacijskog procesa ponovno podijelite populaciju u fronte te ispišite koliko rješenja imate u svakoj fronti. U datoteku "izlaz-dec.txt" ispišite sva pronađena rješenja a u datoteku "izlaz-obj.txt" ispišite njihove dobrote; za problem 2 u ovoj posljednjoj datoteci ispisat ćete uređene parove  $(f_1, f_2)$ . Sadržaj datoteke s dobrotama prikažite grafički (koristite neki gotov alat za vizualizaciju; npr. LibreOffice Draw može generirati graf s ovim podacima). Fiksirajte osi prikaza na 0.1 do 1 za  $f_1$  te na 0 do 10 za  $f_2$ .

Ako Vam za pokretanje programa treba još koja vrijednost koja je specifična za problem koji rješavate, predajte je kao argument komandne linije i to dokumentirajte u datoteci README.txt. Predana zadaća mora u vršnom direktoriju projekta imati i dvije PNG datoteke: jednu koja prikazuje dobiveni rezultat optimizacije problema 2 kada se koristi dijeljenje u prostoru rješenja te drugu koja prikazuje dobiveni rezultat optimizacije problema 2 kada se koristi dijeljenje u prostoru kriterijskih funkcija.

*Napomene:*

Rok za predaju Eclipse projekta je četvrtak, 12. prosinca do 14:00, kako biste mogli pratiti algoritam koji se nadovezuje na ovaj.