

Utjecaja parametara algoritma diferencijalne evolucije na njegovu učinkovitost

Turić, Kristijan

Undergraduate thesis / Završni rad

2025

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:200:449388>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-03-29**

Repository / Repozitorij:

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I
INFORMACIJSKIH TEHNOLOGIJA OSIJEK**

Sveučilišni prijediplomski studij Računarstvo

**Utjecaj parametara algoritma diferencijalne evolucije na
njegovu učinkovitost**

Završni rad

Kristijan Turić

Osijek, 2024.

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK**Obrazac Z1P: Obrazac za ocjenu završnog rada na sveučilišnom prijediplomskom studiju****Ocjena završnog rada na sveučilišnom prijediplomskom studiju**

Ime i prezime pristupnika:	Kristijan Turić
Studij, smjer:	Sveučilišni prijediplomski studij Računarstvo
Mat. br. pristupnika, god.	R4581, 28.07.2020.
JMBAG:	0165086892
Mentor:	doc. dr. sc. Dražen Bajer
Sumentor:	
Sumentor iz tvrtke:	
Naslov završnog rada:	Utjecaja parametara algoritma diferencijalne evolucije na njegovu učinkovitost
Znanstvena grana završnog rada:	Umjetna inteligencija (zn. polje računarstvo)
Zadatak završnog rada:	Opisati algoritam diferencijalne evolucije kao vrstu evolucijskih algoritama s naglaskom na numeričku optimizaciju. Posebno se osvrnuti na parametre algoritma i preporuke u literaturi za njihovo postavljanje. Ugraditi jednu inačicu algoritma za potrebe eksperimentalne analize. Na nekoliko testnih funkcija sustavno ispitati učinkovitost algoritma za različite vrijednosti parametara. Rezervirano za: Kristijan Turić
Datum prijedloga ocjene završnog rada od strane mentora:	18.09.2024.
Prijedlog ocjene završnog rada od strane mentora:	Vrlo dobar (4)
Datum potvrde ocjene završnog rada od strane Odbora:	19.02.2025.
Ocjena završnog rada nakon obrane:	Vrlo dobar (4)
Datum potvrde mentora o predaji konačne verzije završnog rada čime je pristupnik završio sveučilišni prijediplomski studij:	20.02.2025.



FERIT

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

IZJAVA O IZVORNOSTI RADA

Osijek, 20.02.2025.

Ime i prezime Pristupnika:

Kristijan Turić

Studij:

Sveučilišni prijediplomski studij Računarstvo

Mat. br. Pristupnika, godina upisa:

R4581, 28.07.2020.

Turnitin podudaranje [%]:

8

Ovom izjavom izjavljujem da je rad pod nazivom: **Utjecaja parametara algoritma diferencijalne evolucije na njegovu učinkovitost**

izrađen pod vodstvom mentora doc. dr. sc. Dražen Bajer

i sumentora

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.

Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis pristupnika:

SADRŽAJ

1. UVOD	1
1.1. Zadatak završnog rada	1
2. ALGORITAM DIFERENCIJALNE EVOLUCIJE I NJEGOVI PARAMETRI	2
2.1. Struktura algoritma	3
2.2. Parametri algoritma	6
2.3. Prijedlozi postavki parametara u literaturi.....	7
3. OSTVARENO PROGRAMSKO RJEŠENJE	9
3.1. Način rada programskog rješenja	10
3.2. Prikaz i način uporabe programskog rješenja	11
4. EKSPERIMENTALNA ANALIZA.....	13
4.1. Postavke eksperimenta	14
4.2. Rezultati	14
5. ZAKLJUČAK.....	21
LITERATURA	22
SAŽETAK.....	23
ŽIVOTOPIS.....	24

1. UVOD

Kontinuirana optimizacija je problem koji se često javlja u različitim inženjerskim zadacima, kao što su dizajn, modeliranje i simulacija, a posebno je prisutna i u strojnom učenju. Glavni cilj kontinuirane optimizacije je pronaći optimalna rješenja u prostoru rješenja koji je neprekidan, što znači da varijable mogu poprimiti bilo koju vrijednost unutar zadanih granica. Međutim, ovi problemi nose sa sobom niz izazova, poput složenosti ciljne funkcije, funkcija nema derivaciju ili je isprekidana. Isto tako funkcija često može biti višemodalna ili bez poznatog analitičkog oblika.

Evolucijski algoritmi su valjani postupci za rješavanje ovakvih problema jer ne zahtijevaju informacije o derivaciji funkcije. Evolucijski algoritmi su inspirirani procesima prirodne evolucije i selekcije, te uključuju optimizacijske postupke koji su uspješni u traženju rješenja u kompleksnim prostorima. Algoritam diferencijalne evolucije je posebno značajan predstavnik evolucijskih algoritama, koji je pokazao dobre rezultate u rješavanju problema kontinuirane optimizacije. Ipak, njegova učinkovitost, kao i kod drugih evolucijskih algoritama, osjetljiva je na postavke parametara, kao što su veličina populacije, faktor skaliranja i stopa križanja. Nema univerzalno dobrih postavki za sve probleme, pa je potrebna prilagodba ovisno o prirodi problema. U literaturi postoje razne preporuke za početne postavke parametara, koje su pokazale dobre rezultate na raznim problemima. Te postavke moguće je koristiti ili iskoristi za traženje boljih postavki parametara.

Drugo poglavlje daje pregled algoritma diferencijalne evolucije, njegove strukture i glavnih komponenti kao što su mutacija, križanje i selekcija. Detaljno su objašnjeni parametri algoritma, te prijedlozi za njihovo postavljanje u literaturi. U trećem poglavlju opisano je ostvareno programsko rješenje, te način rada samog programskog rješenja. Isto tako prikazan je način uporabe programskog rješenja iz perspektive korisnika. Četvrto poglavlje opisuje postavke eksperimentalne analize, cilj analize, te su prikazani i opisani rezultati eksperimenta.

1.1. Zadatak završnog rada

Opisati algoritam diferencijalne evolucije kao vrstu evolucijskih algoritama s naglaskom na numeričku optimizaciju. Posebno se osvrnuti na parametre algoritma i preporuke u literaturi za njihovo postavljanje. Ugraditi jednu inačicu algoritma za potrebe eksperimentalne analize. Na nekoliko testnih funkcija sustavno ispitati učinkovitost algoritma za različite vrijednosti parametara.

2. ALGORITAM DIFERENCIJALNE EVOLUCIJE I NJEGOVI PARAMETRI

Kontinuirana optimizacija predstavlja proces traženja optimalnog rješenja u prostoru realnih brojeva za probleme u kojima se koriste kontinuirane varijable, odnosno varijable koje mogu poprimiti bilo koju vrijednost unutar zadanih granica. Algoritam diferencijalne evolucije koristi se za rješavanje takvih problema. Problem kontinuirane optimizacije sastoji se od funkcije cilja, nezavisnih varijabli i skupa ograničenja [1]. Funkcija cilja označava se s f . Potrebno je pronaći rješenje koje će maksimizirati ili minimizirati f . Nezavisne varijable su varijable čija se vrijednost može mijenjati kako bi se pronašlo najbolje rješenje za određeni problem. Na primjer, u problemu optimizacije proizvodnje, nezavisne varijable mogu biti količina materijala, vrijeme rada ili brzina proizvodnje. Skup ograničenja su pravila i uvjeti koje rješenje mora zadovoljiti kako bi bilo prihvaćeno. Ako su rješenja ograničena samo granicama prostora pretrage, onda se takav problem naziva neograničenim problemom. Ograničeni problemi imaju dodatna ograničenja. Primjeri neograničenih problema optimizacije su optimizacija dizajna rešetkastih nosača (engl. *truss*) [2], optimizacija razašiljanja električne energije [3], modeliranje i optimizacija toka prometa [4], poboljšanje kvalitete hitne medicinske pomoći [5], podešavanje parametara modela strojnog učenja, te optimizacija težina u umjetnim neuronskim mrežama.

Algoritam optimizacije dodjeljuje vrijednosti nezavisnim varijablama na način da je funkcija cilja optimizirana i da su sva ograničenja zadovoljena. Funkcija je optimizirana kada rješenje dostiže minimum ili maksimum funkcije. Diferencijalna evolucija je algoritam optimizacije korišten za rješavanje problema kontinuirane optimizacije [6], odnosno za probleme gdje su nezavisne varijable predstavljene realnim brojevima [7], [8]. Evolucijski algoritmi temelje se na biološkoj evoluciji i koriste se mehanizmi kao što su mutacija, križanje i selekcija kako bi se iterativnim postupkom poboljšala rješenja. Dakle, ponavljanjem korištenjem operacija mutacije, križanja i selekcije dobivaju se sve bolja rješenja. Algoritam DE temelji se na evolucijskim algoritmima [9].

Standardna verzija algoritma diferencijalne evolucije koristi tri parametra o kojima ovisi učinkovitost algoritma. Ti parametri su veličina populacije (NP), faktor skaliranja (F) i stopa križanja (CR). Zato što o njima ovisi učinkovitost postoji mnogo preporuka i načina za njihovo podešavanje u literaturi.

2.1. Struktura algoritma

Algoritam diferencijalne evolucije sastoji se od populacije rješenja, operatora (mutacija, križanje i selekcija), funkcije cilja, parametara i uvjeta završetka. Populacija rješenja označava slijed vektora, gdje svaki vektor predstavlja jedno potencijalno rješenje problema koji se pokušava riješiti. Prikaz pseudo-kôdom algoritma DE dan je na slici 2.1.

Linija Kod

```
1:   Postaviti vrijednosti parametara NP, F i CR
2:   Inicijalizirati populaciju P
3:   Dok uvjet završetka nije ispunjen:
4:     Za svaki vektor  $\mathbf{v}^{j,g}$  u populaciji P
5:       Odaberi nasumično  $\mathbf{v}^{r1}, \mathbf{v}^{r2}, \mathbf{v}^{r3}$  različite od j
6:       Generiraj mutant vektor  $\mathbf{u}^{j,g}$ 
7:       Napravi križanje vektora  $\mathbf{u}^{j,g}$  i  $\mathbf{v}^{j,g}$  da bi stvorio  $\mathbf{t}^{j,g}$ 
8:       Ako  $f(\mathbf{t}^{j,g}) \leq f(\mathbf{v}^{j,g})$ :
9:          $\mathbf{t}^{j,g}$  zamjenjuje  $\mathbf{v}^{j,g}$  u populaciji
10:  Kraj
```

Sl. 2.1. Pregled algoritma DE

Inicijalizacija populacije rješenja prvi je korak pri izvođenju algoritma DE. Sama rješenja u populaciji nazivaju se vektorima. Vektor je određen svojom dimenzijom, te vrijednostima od kojih se sastoji. Označava ga se s \mathbf{v}^j , gdje j označava indeks vektora u populaciji, a \mathbf{v}_i^j označava komponentu vektora s indeksom i [10]. Prema tome, i će se nalaziti u intervalu od 0 do d , gdje je d dimenzija samog vektora. Kod populacije rješenja bitna je vrijednost na koju je postavljen parametar NP jer on određuje broj vektora koji se inicijalizira. Populacija se označava s $P = (\mathbf{v}^{1,g}, \dots, \mathbf{v}^{NP,g})$ [11]. Oznaka g odnosi se na generaciju vektora u populaciji. Na taj način može se prikazati koliko je generacija neka populacija prošla ili usporediti vektore prošle i trenutačne generacije. Postupak inicijalizacije populacije sastoji se od stvaranja vektora unutar granica prostora pretrage. Granice prostora pretrage ovise o problemu koji se pokušava riješiti, odnosno o funkciji cilja koja se koristi za vrednovanje vektora. Vrijednost svake komponente vektora postaviti će se na nasumičnu vrijednost unutar granica prostora pretrage. Dodjeljivanje vrijednosti komponenti i vektora j iz populacije P može se prikazati formulom [7]:

$$\mathbf{v}_i^j = \min + U_i \times (\max - \min), \quad (2-1)$$

gdje su \min i \max vektori unutar \mathbb{R}^d koji označavaju donju i gornju granicu prostora pretrage za svaku komponentu vektora. U_i je nasumično uniformno generiran broj od 0 do 1. Kada je broj

nasumično uniformno generiran, to znači da je jednako vjerojatno dobiti bilo koji broj iz tog intervala kao rezultat.

Jedna od ključnih elemenata algoritma diferencijalne evolucije je mutacija, po kojoj je algoritam dobio ime [7]. Mutacija je postupak kojim se dobiva novi vektor koristeći vektore trenutačne populacije. Vektor koji je rezultat mutacije označen je s \mathbf{u}^j i naziva se mutant vektor. Kako bi se proveo postupak mutacije potrebno je nasumično izabrati tri vektora iz populacije, koja su različita od trenutačno odabranog vektora (\mathbf{v}^j) iz populacije. Nakon toga, razlika dvaju vektora množi se s faktorom skaliranja F , te se zbraja s preostalim vektorom. Za svaki vektor član populacije stvara se mutant vektor \mathbf{u}^j , dakle stvara se NP mutant vektora. Nasumično izabrani vektori koji sudjeluju u postupku mutacije označavaju se s \mathbf{v}^{r1} , \mathbf{v}^{r2} i \mathbf{v}^{r3} . Indeksi $r1$, $r2$, $r3$ i j , ne smiju biti jednaki. Dakle, to se može prikazati relacijom $j \neq r1 \neq r2 \neq r3$. Opisana strategija mutacije naziva se rand/1 strategija [11]. Postoje i druge strategije koje se koriste ovisno o problemu koji se rješava. Mutacija je glavni element o kojem ovisi učinkovitost algoritma, zato što daljnji postupci u algoritmu ovise o stvorenim vektorima. Prikaz rand/1 strategije u obliku formule [11]:

$$\mathbf{u}^j = \mathbf{v}^{r1} + F \cdot (\mathbf{v}^{r2} - \mathbf{v}^{r3}) \quad (2-2)$$

Zbog operatora mutacije vrijednosti novonastalih komponenti mutant vektora \mathbf{u}^j mogu izaći iz granica prostora pretrage. Tijekom procesa mutacije razlika vektora množi se faktorom skaliranja, te se onda to nadodaje na vektor \mathbf{v}^{r1} . Pri tome vrijednost komponente vektora može izaći izvan granica prostora pretrage. Rukovanje tim komponenta bitan je dio algoritma diferencijalne evolucije [11]. Jedan od načina rukovanja tim vrijednostima je postavljanje vrijednosti koja je izašla iz granice na vrijednost najbliže granice prostora pretrage kao što je navedeno u [11]. Ovaj postupak može se prikazati formulom:

$$\mathbf{u}_i^j = \begin{cases} \min, \text{ ako je } \mathbf{u}_i^j < \min \\ \max, \text{ ako je } \mathbf{u}_i^j > \max \end{cases} \quad (2-3)$$

Postoje i drugi načini rukovanja vrijednostima koje su izašle iz granica kao što je navedeno u [8].

Nastali mutant vektor \mathbf{u}^j križa se s trenutačno odabranim vektorom (\mathbf{v}^j). Postupkom križanja nasljeđuju se komponente vektora \mathbf{u}^j i vektora \mathbf{v}^j , te se stvara novi probni vektor \mathbf{t}^j . Za svaku komponentu vektora \mathbf{t}^j radi se odabir između komponenti vektora \mathbf{u}^j i \mathbf{v}^j . Postupak odabira komponente dan je kao [11]:

$$\mathbf{t}_i^j = \begin{cases} \mathbf{u}_i^j, & \text{ako } U_i \leq CR \vee i = r_j, \\ \mathbf{v}_i^j, & \text{inače,} \end{cases} \quad i = 1, \dots, d \quad (2-4)$$

Dakle, ovisno o vrijednosti koja je nasumično dobivena i vrijednosti parametra CR odabire se komponenta. Dio jednadžbe $i = r_j$ osigurava pojavljivanje barem jedne komponente iz mutant vektora u probnom vektoru \mathbf{t}^j . r_j je nasumično izabran broj u rasponu $\{0, \dots, d\}$. Križanje omogućuje stvaranje novih vektora iz postojećih vektora kombiniranjem njihovih komponenti. Time je moguće dobiti bolji vektor, odnosno bolje rješenja problema. Ovaj operator križanja naziva se binomno križanje, te je označena s *bin* [11].

Zadnji korak u algoritmu diferencijalne evolucije je odabir između vektora \mathbf{v}^j i \mathbf{t}^j . Uspoređuju se vrijednosti ciljne funkcije za vektore \mathbf{v}^j i \mathbf{t}^j . Ako je vrijednost ciljne funkcije vektora \mathbf{t}^j manja ili jednaka od vrijednosti za \mathbf{v}^j , onda će se vektor \mathbf{v}^j zamijeniti s vektorom \mathbf{t}^j u populaciji. Time je osigurano da je svaka iduća generacija populacije bolja ili u najgorem slučaju jednako dobra kao i prošla. Postupak selekcije može se prikazati formulom [11]:

$$\mathbf{v}^{j,g+1} = \begin{cases} \mathbf{t}^{j,g}, & \text{ako } f(\mathbf{t}^{j,g}) \leq f(\mathbf{v}^{j,g}) \\ \mathbf{v}^{j,g}, & \text{inače.} \end{cases} \quad (2-5)$$

U navedenoj formuli $g+1$ označava iduću generaciju populacije dok g označava trenutačnu generaciju populacije.

Početna populacija inicijalizira se jednom, te se nakon toga ponavlja proces primjene operatora, mutacije i križanja, nakon kojih se izabiru bolji vektori s pomoću selekcije sve dok nije postignut uvjet završetka samog algoritma. Uvjet je postavljen ovisno o problemu koji se rješava. U ovom radu uspoređuju se drugačije postavke parametara algoritma, te se treba osigurati pošteni usporedba istih. Prema tome za uvjet završetka algoritma uzima se maksimalni broj evaluacija funkcije cilja označen s $NFES_{max}$. Korištenje $NFES_{max}$ kao uvjeta završetka omogućuje da se svakoj postavci parametara dodijeli isti broj evaluacija funkcije cilja. Kada bi se koristio broj iteracija kao uvjet završetka, postavke parametara s većom veličinom populacije imale bi više vrednovanja funkcije cilja od onih s manjom veličinom populacije. Iz tog razloga korištenje $NFES_{max}$ daje ravnopravniju usporedbu postavki parametara. Na taj način uspoređuju se kvalitete rješenja unutar istog broja evaluacija. $NFES_{max}$ je cijeli broj koji se postavlja prije provođenja eksperimenta.

2.2. Parametri algoritma

Učinkovitost algoritma vrlo je ovisna o postavkama parametara algoritma [14]. Stoga je odabir odgovarajućih postavki bitan kako bi dobili bolja rješenja. Tri su glavna parametra koja se postavljaju prije izvođenja algoritma, veličina populacije, faktor skaliranja i stopa križanja. Svaki problem zahtjeva drugačije postavke. Parametri se mogu postaviti prije izvođenja samog algoritma, ali mogu se mijenjati i tijekom izvođenja algoritma [14]. U ovom radu postavke koje se uspoređuju se postavljaju prije izvođenja algoritma, te se drže fiksima tijekom izvođenja.

Veličina populacije se označava s NP , te izravno utječe na broj vektora koji će biti inicijalizirani tijekom postupka inicijalizacije populacije. Veće vrijednosti veličine populacije bolje pokrivaju prostor pretrage. Svaki vektor inicijaliziran je unutar granica prostora pretrage, te što je više vektora inicijalizirano to je više samog prostora istraženo [14]. To isto tako znači da populacija sadrži više mogućih rješenja. Nedostatak velikih vrijednosti za NP je povećana računalna složenost, čime se povećava vrijeme potrebno da algoritam dođe do dobrih rješenja. Za svaki vektor primjenjuju se operatori mutacije i križanja, te se poziva funkcija cilja. Kod manjih vrijednosti veličine populacije moguće je ne imati dovoljno članova populacije kako bi se došlo do optimalnih rješenja. Ako se to dogodi rješenja u populaciji ne postižu optimalnu vrijednost već se nalaze unutar ograničenog raspona blizu početne populacije [14]. To se naziva stagnacija [14]. Pojava stagnacije ukazuje na nepravilno postavljene parametre. Zbog toga bitno je odabrati veličinu populacije koja ima dovoljno članova populacije kako bi se prostor pretrage istražio bez rizika stagnacije algoritma.

Faktor skaliranja određuje veličinu promjene prilikom mutacije u diferencijalnoj evoluciji. Tijekom mutacije, razlika između dva vektora množi se s faktorom skaliranja. Veće vrijednosti faktora skaliranja povećavaju razliku dva vektora, a time i promjenu u vektoru na koji se nadodaje razlika (\mathbf{v}^{r1}). Time se postiže veće istraživanje prostora pretrage. Stvaraju se vektori koji više odstupaju od vektora \mathbf{v}^{r1} . Efektivni raspon faktora skaliranja je $< 0, 1$] [14]. Postavljanje faktora skaliranja na 0, isključuje sam proces mutacije iz algoritma, stoga ga nema smisla koristiti. Manje vrijednosti faktora skaliranja uvode manju promjenu za vektor \mathbf{v}^{r1} . Na taj način manje istražujemo prostor pretrage, ali se istražuje prostor u blizini vektora \mathbf{v}^{r1} .

Stopa križanja određuje vjerojatnost odabira komponente mutant vektora \mathbf{u}^j naspram komponente vektora \mathbf{v}^j . Zato što se stopa križanja uspoređuje s brojem U_i , čija je vrijednost od 0 do 1, stopa križanja isto tako mora biti u intervalu od 0 do 1. Niže vrijednosti stope križanja smanjuju udio komponenti mutant vektora u probnom vektoru \mathbf{t}^j . Ako se vrijednost stope križanja postavi na

nulu, probni vektor imat će samo jednu komponentu mutant vektora, koja je dana s r_j . Veće vrijednosti stope križanja povećavaju omjer komponenti mutant vektora u probnom vektoru, te ako se postavi na 1 probni vektor imat će sve komponente mutant vektora. Probni vektor koji ima više komponenti iz mutant vektora više će istražiti prostor pretrage izvan postojeće populacije. Time se dobiva više mogućih rješenja.

Različiti problemi zahtijevaju drugačije postavke navedenih parametara [15]. U literaturi se često naglašava potreba za prilagodbom parametara ovisno o problemu, što uključuje postavljanje parametara prije izvođenja algoritma i postavljanje parametara tijekom izvođenja algoritma [15].

2.3. Prijedlozi postavki parametara u literaturi

Prikladne postavke potrebne su kako bi se dobilo optimalno rješenje problema koji se pokušava riješiti. Traženje prikladnih postavki je proces koji zahtjeva mnogo vremena i računalnih resursa [16]. Zbog toga se pri pokušaju rješavanja problema i traženja dobrih postavki za dan problem obično koriste prijedlozi dani u literaturi. Time dajući dobar temelj za daljnje podešavanje parametara neovisno o tome podešavaju li se parametri ručno prije izvođenja algoritma ili adaptivno tijekom izvođenja algoritma. Razne preporuke postoje za svaki od parametara algoritma. Važno je napomenuti da su parametri ovisni jedan o drugome, te promjena jednog parametra može zahtijevati i promjenu drugih parametara [14]. Ne postoji univerzalno dobar algoritam za sve probleme, odnosno potrebno je adaptirati algoritam ovisno o problemu kako bi se poboljšala efikasnost i pronašla dobra rješenja [16].

Veličina populacije u većini slučajeva postavlja se na vrijednost od $5d$ do $10d$ [6]. Time, veličina populacije ovisi o dimenziji d . Cilj dobrog postavljanja ovog parametra je imati dovoljno potencijalnih rješenja kako bi se dostigao optimum funkcije, ali ne imati previše rješenja zbog vremena potrebnog da algoritam dođe do dobrih rješenja. U [2] veličina populacije postavljenja je na $10d$, čime se više istražio prostor pretrage. Donja granica intervala, $5d$, korištena je u [12], gdje su dobra rješenja dobivena brže zbog smanjene računalne složenosti, odnosno manje populacije. Kod problema s većim brojem dimenzija najčešće se koriste vrijednosti 50 ili 100 [14], kako bi se smanjila računalna složenost i izbjegla ovisnost o dimenziji.

Faktor skaliranja obično se postavlja na 0.5 [14], ali vrijednosti u intervalu [0.4, 1.0] mogu biti korisne ovisno o problemu [2], [13]. U [6] se preporučuje 0.5 kao zadana početna vrijednost, koja se pokazala učinkovita za većinu optimizacijskih problema.

Vrijednost stope križanja najčešće se postavlja na 0.9 [14] ili na vrijednost iz intervala [0.5, 0.9]. Za većinu problemu u [6] predložena je vrijednost 0.9, ista se koristila i u [12] kao zadana postavka. U [15] korištena je vrijednost 0.6 za stopu križanja. Vrijednost od 0.8 korištena je u eksperimentima [2]. Najčešće korištene preporučene vrijednosti prikazane su u tablici 2.1.

Tablica 2.1. Tablični pregled preporuka

<i>Parametar</i>	<i>Preporučena vrijednost</i>	<i>Literatura</i>
<i>Veličina populacije (NP)</i>	100, 50 ili [5 <i>d</i> , 10 <i>d</i>]	[2], [6], [14]
<i>Faktor skaliranja (F)</i>	0.5 ili [0.4, 1.0]	[6], [12]
<i>Stopa križanja (CR)</i>	0.9 ili [0.5, 0.9]	[13], [14], [15]

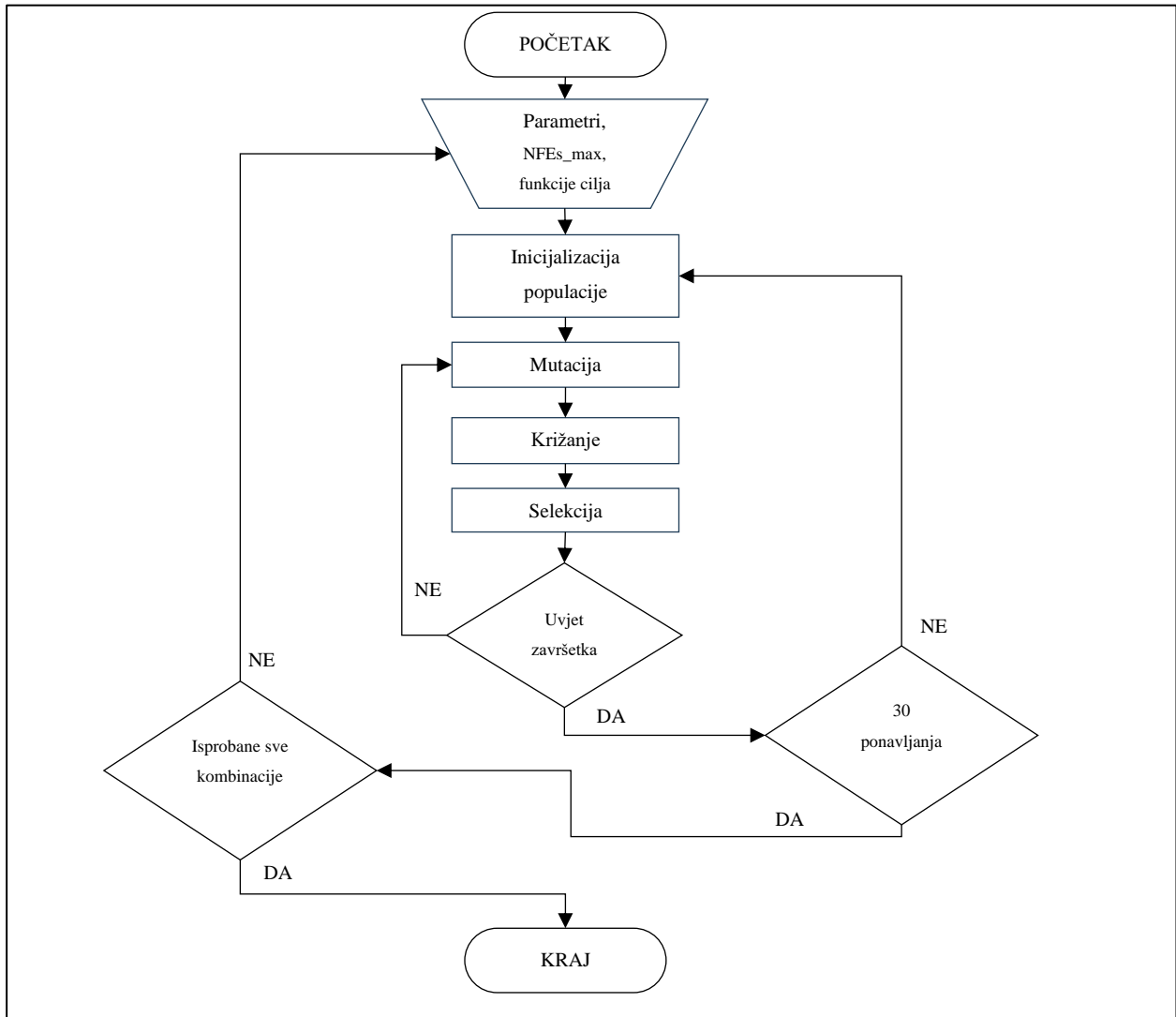
3. OSTVARENO PROGRAMSKO RJEŠENJE

Ostvareno programsko rješenje uključuje implementaciju diferencijalne evolucije s mogućnošću postavljanja različitih parametara algoritma, kao što su veličina populacija, faktor skaliranja i stopa križanja. Rješenje omogućuje definiranje uvjeta završetka izvođenja algoritma, odnosno postavljanje maksimalnog broja vrednovanja funkcije cilja ($NFE_{s_{max}}$). Isto tako, omogućava odabir funkcije cilja s pomoću koje se evaluira kombinacija parametara, te pruža korisniku rezultate u obliku tablice. Rezultati sadrže najbolje postignute vrijednosti funkcije tijekom izvođenja algoritma.

Pored toga, programsko rješenje omogućuje praćenje kvalitete najboljeg rješenja kroz nekoliko ključnih točaka tijekom izvođenja algoritma. Na taj način se pri crtanju grafa dobiva uvid u promjenu najbolje vrijednosti funkcije cilja. Implementacija također olakšava izvođenje eksperimenta te prikupljanje i analizu podataka, pružajući uvid u učinkovitost algoritma ovisno o različitim postavkama parametara. Ostvaren je automatski sustav provođenja eksperimenta, gdje se jednim pokretanjem programa dobivaju rezultati svih postavki parametara u formatu koji se može prikazati u obliku tablice. Nakon toga podatci iz tablice prikazuju se s pomoću grafova.

3.1. Način rada programskog rješenja

Ostvareno programsko rješenje diferencijalne evolucije u obliku dijagrama toka prikazano je na slici 3.1.



Sl. 3.1. Dijagram toka programskog rješenja

Prvi korak je postavljanje kombinacija parametara, postavljanje dimenzija, te postavljanje funkcija koje će se koristiti za evaluaciju. Isto tako postavlja se vrijednost $NFEs_{max}$. Nakon što je sve postavljeno pokreće se programsko rješenje.

Inicijalizira se početna populacija, te se ponavljaju operacije mutacije, križanja i selekcije sve dok nije ispunjen uvjet završetka algoritma, odnosno sve dok nije postignut $NFEs_{max}$ broj vrednovanja funkcije cilja. Nakon što se postigne $NFEs_{max}$ i algoritam diferencijalne evolucije završi s izvođenjem, rezultati se zapisuju u datoteku. Ovaj postupak ponavlja se za svaku od izabranih kombinacija parametara, te ponovno za svaku funkciju cilja. Isto tako, ponavlja se i za svaku od dvije dimenzije koje su zadane. Važno je napomenuti da se zbog bolje usporedbe i

računanja deskriptivne statistike svaka pojedina kombinacija parametara, funkcije i dimenzija ponavlja 30 puta. Ostvareno programsko rješenje zapisuje rezultate svake iteracije algoritma. Sami algoritam DE, kako je navedeno u prethodnim poglavljima, zaustavlja se nakon što je postignuto $NFEs_{max}$ poziva funkcije cilja. Program zapisuje rezultate na 1%, 2%, 3%, 5%, 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90% i 100% vrijednosti $NFEs_{max}$. Zapisani rezultati uključuju ime funkcije, veličinu populacije, faktor skaliranja, dimenziju, postotak $NFEs_{max}$ do kojeg je algoritam došao, te pogrešku. Pogreška se s vremenom smanjuje, te time se dobiva uvid u stopu konvergencije svake pojedine postavke parametara. Nakon dobivanja rezultata u obliku tablice, moguć je prikaz u obliku grafova.

3.2. Prikaz i način uporabe programskog rješenja

Ostvareno programsko rješenje omogućuje jednostavno provođenje eksperimenta na algoritmu diferencijalne evolucije. Glavna cilj programa je ispitivanje efikasnosti različitih postavki parametara. U nastavku je opisan način uporabe programa.

Korisnik prvo postavlja kombinacije parametara koje želi ispitati. U ovom slučaju program radi s deset kombinacija parametara. Nakon toga postavljaju se vrijednosti za dimenzije koje se žele koristiti, te vrijednost $NFEs_{max}$. Na kraju, izabiru se funkcije cilja koje se koriste. U programu implementirano je šest funkcija cilja, te se sve koriste pri evaluaciji. Kada je sve postavljano kreće provođenje eksperimenta. Postupak je automatski, te korisnik samo dobiva povratnu informaciju o tome koja koji dio eksperimenta je gotov, kako bi mogao pratiti napredak. Taj prikaz vidljiv je na slici 3.2.

```
Pop size: 100
Mutation factor: 0.50
Crossover rate: 0.90
Dimension: 30
Function: Spherical Function

Pop size: 100
Mutation factor: 0.50
Crossover rate: 0.30
Dimension: 30
Function: Spherical Function
|
```

Sl. 3.2. Prikaz korisnikovog sučelja tijekom izvođenja

Nakon što je program gotov s izvođenjem, korisnik dobiva datoteka u kojoj su zapisani svi podatci potrebni za daljnju analizu i usporedbu kombinacija parametara. Način na koji su podatci prikazani može se vidjeti na slici 3.3.

function	NP	F	CR	D	NFEmax	fxerror
Spherical	100	0.5	0.9	30	0.01	28090.583150485123042
Spherical	100	0.5	0.9	30	0.02	14654.744228910491074
Spherical	100	0.5	0.9	30	0.03	4633.780016485326996
Spherical	100	0.5	0.9	30	0.05	687.372812221277286
Spherical	100	0.5	0.9	30	0.1	11.797524557336232
Spherical	100	0.5	0.9	30	0.2	0.003925
Spherical	100	0.5	0.9	30	0.3	1.68E-06
Spherical	100	0.5	0.9	30	0.4	8.74E-10
Spherical	100	0.5	0.9	30	0.5	1.49E-13
Spherical	100	0.5	0.9	30	0.6	0
Spherical	100	0.5	0.9	30	0.7	0
Spherical	100	0.5	0.9	30	0.8	0
Spherical	100	0.5	0.9	30	0.9	0
Spherical	100	0.5	0.9	30	1	0
Spherical	100	0.5	0.9	30	0.01	21779.458642695746676
Spherical	100	0.5	0.9	30	0.02	12193.267982739893341
Spherical	100	0.5	0.9	30	0.03	4576.623874890444313
Spherical	100	0.5	0.9	30	0.05	903.355462605965045
Spherical	100	0.5	0.9	30	0.1	21.065349804451483
Spherical	100	0.5	0.9	30	0.2	0.006268
Spherical	100	0.5	0.9	30	0.3	1.97E-06
Spherical	100	0.5	0.9	30	0.4	2.86E-10
Spherical	100	0.5	0.9	30	0.5	9.8E-14

Sl. 3.3. Prikaz dijela dobivenih rezultata

Podatci se, pokretanjem drugog dijela programskog rješenja, mogu automatski obraditi i prikazati u obliku grafova. To znači da ostvareno programsko rješenje nudi mogućnost grafičkog prikaza.

4. EKSPERIMENTALNA ANALIZA

Cilj eksperimentalne analize je ispitati ponašanje i učinkovitost algoritma diferencijalne evolucije pri različitim postavkama parametara. Analiza se provodi kako bi se utvrdilo kako promjena ključnih parametara, veličine populacije, faktora skaliranja i stope križanja, utječe na performanse algoritma ovisno o određenom problemu odnosno funkciji cilja. U ovom radu ispitano je šest funkcija cilja, od kojih tri su unimodalne, a tri su multimodalne. Unimodalne funkcije su funkcije koje imaju samo globalni minimum ili maksimum, dok multimodalne funkcije imaju više lokalnih minimuma ili maksimuma. Kod unimodalnih funkcija lakše je doći do optimuma funkcije. Korištene su tri unimodalne funkcije, Sferna funkcija, Rosenbrockova funkcija i Zakharova funkcija, te tri multimodalne funkcije, Schwefelova 2.26 funkcija, Rastriginova funkcija i Ackleyeva funkcija. Korištene funkcije i njihove karakteristike navedene su u tablici 4.1.

Tablica 4.1. Funkcije korištene pri eksperimentalnoj analizi

<i>Funkcija</i>	<i>Dimenzionalnost (d)</i>	<i>Prostor pretrage</i>	<i>Optimum</i>	<i>Vrijednost u optimumu</i>
<i>Sferna funkcija</i>	$\{30, 50\}$	$[-100, 100]^d$	$\{0, 0, \dots, 0\}$	0
<i>Rosenbrockova funkcija</i>	$\{30, 50\}$	$[-10, 10]^d$	$\{1, 1, \dots, 1\}$	0
<i>Zakharova funkcija</i>	$\{30, 50\}$	$[-10, 10]^d$	$\{0, 0, \dots, 0\}$	0
<i>Schwefelova funkcija</i>	$\{30, 50\}$	$[-500, 500]^d$	$\{c, c, \dots, c\}$ $c = 420.96874\dots$	0
<i>Rastriginova funkcija</i>	$\{30, 50\}$	$[-5.12, 5.12]^d$	$\{0, 0, \dots, 0\}$	0
<i>Ackleyeva funkcija</i>	$\{30, 50\}$	$[-32.768, 32.768]^d$	$\{0, 0, \dots, 0\}$	0

U ovom radu korištene su dimenzije 30 i 50 za svaku kombinaciju parametara. Odabrano je deset od nekoliko najčešće korištenih kombinacija parametara [14]. Kombinacije su odabrane kako bi se mogle usporediti, te promatrati kako promjena njihovih vrijednosti utječe na rezultate eksperimenta. Na primjer, prve dvije kombinacije parametara su iste, osim što je stopa križanja značajno drugačija. Na ovaj način može se vidjeti utjecaj promjene tog parametra na rezultate. Postavke parametara koje su ispitane u ovom radu navedene su u tablici 4.2.

Tablica 4.2. Ispitane kombinacije parametara

<i>Veličina populacije (NP)</i>	<i>Faktor skaliranja (F)</i>	<i>Stopa križanja (CR)</i>
100	0.5	0.9
100	0.5	0.3
100	0.8	0.6
50	0.5	0.9
50	0.9	0.9
50	0.9	0.1
30	0.5	0.9
30	0.5	0.1
10d	0.5	0.9
10d	0.5	0.5

4.1. Postavke eksperimenta

Eksperiment se sastoji od dvije dimenzije, deset kombinacija parametara i šest funkcija cilja. Tijekom provođenja eksperimenta ispituju se sve kombinacije dimenzija, parametara i funkcija cilja. Kako je ranije navedeno, svaka pojedina kombinacija izvodi se dok ne dostigne $NFE_{s_{max}} = d * 10^4$ evaluacija funkcije cilja. Osim toga, potrebno je višestruko izvođenje svake pojedine kombinacije kako bi se dobio konzistentan uvid u rezultate. U ovom radu svaka pojedina kombinacija ponavljanja je 30 puta, zbog stohastičke prirode algoritma DE. Višestruka ponavljanja ključna su za dobivanje uvida u varijabilnost rezultata i za prosječnu kvalitetu rješenja. Dakle, u jednom provođenju eksperimenta postoji 120 kombinacija dimenzije, parametara i funkcije cilja, te ako ponovimo svaku od tih kombinacija 30 puta. To je sveukupno 3600 izvođenja algoritma diferencijalne evolucije.

Tijekom jednog izvođenja algoritma DE prikupljeno je 14 vrijednosti, koje se zapisuju u datoteku. Tih 14 vrijednosti odnosi se na vrijednost najmanje pogreške, odnosno to je razlika vrijednosti funkcije cilja trenutačno najboljeg rješenja u populaciji i optimalne vrijednosti funkcije cilja. Nakon prikupljanja svih rezultata, može se izračunati deskriptivna statistika. U ovom slučaju, zbog 30 ponavljanja istih postavki, moguće je dobiti prosječnu vrijednost, te standardnu devijaciju. Korištenjem deskriptivne statistike i bilježenjem rezultata omogućena je usporedba učinkovitosti i stabilnosti DE algoritma ovisno o različitim postavkama parametara.

4.2. Rezultati

Eksperimentalna analiza provedena je kako bi se ispitalo ponašanje i učinkovitost algoritma diferencijalne evolucije pri različitim postavkama parametara. Rezultati su prikazani u obliku grafova, te u obliku tablica koje prikazuju njihove prosječne vrijednosti i standardne devijacije. U tablicama prikazane su srednje vrijednosti rezultata i standardna devijacija ovisno o parametrima, funkcijama cilja, te dimenzijama.

Tablica 4.3. Rezultati postavki parametara ovisno o funkcijama sa $d=30$

NP	F	CR	Sferna funkcija	Rosenbrockova funkcija	Zakharova funkcija	Schwefelova funkcija	Rastriginova funkcija	Ackleyeva funkcija
100	0.5	0.9	0.000e+00	9.476e-01	0.000e+00	4.156e+03	1.308e+02	4.900e-15
100	0.5	0.3	0.000e+00	2.230e+01	0.000e+00	3.577e-12	4.881e+01	6.300e-15
100	0.8	0.6	4.269e-01	3.274e+02	5.079e+00	4.692e+03	1.860e+02	4.843e-01
50	0.5	0.9	0.000e+00	2.619e+01	0.000e+00	2.447e+03	1.439e+01	5.900e-15
50	0.9	0.9	9.470e-03	2.579e+01	2.231e-02	2.416e+03	6.138e+01	3.039e-02
50	0.9	0.1	0.000e+00	2.526e+01	0.000e+00	3.948e+00	0.000e+00	1.240e-14
30	0.5	0.9	5.325e+00	5.480e+01	1.606e+03	2.965e+03	2.227e+01	1.100e+00
30	0.5	0.1	0.000e+00	2.781e+01	0.000e+00	7.501e+01	0.000e+00	7.000e-15
10d	0.5	0.9	1.241e-03	2.358e+01	1.828e-03	6.974e+03	1.944e+02	1.389e-02
10d	0.5	0.5	4.840e-09	2.457e+01	3.800e-08	5.456e+03	1.272e+02	2.018e-05

Tablica 4.4. Standardne devijacije rezultata postavki parametara ovisno o funkcijama sa $d=30$

NP	F	CR	Sferna funkcija	Rosenbrockova funkcija	Zakharova funkcija	Schwefelova funkcija	Rastriginova funkcija	Ackleyeva funkcija
100	0.5	0.9	\pm 0.00e+00	\pm 7.96e-01	\pm 0.00e+00	\pm 1.08e+03	\pm 2.15e+01	\pm 1.37e-15
100	0.5	0.3	\pm 0.00e+00	\pm 4.32e-01	\pm 0.00e+00	\pm 3.27e-13	\pm 4.31e+00	\pm 1.27e-15
100	0.8	0.6	\pm 1.21e-01	\pm 6.49e+01	\pm 2.11e+00	\pm 2.24e+02	\pm 1.23e+01	\pm 7.98e-02
50	0.5	0.9	\pm 0.00e+00	\pm 1.44e+01	\pm 0.00e+00	\pm 5.96e+02	\pm 3.53e+00	\pm 1.45e-15
50	0.9	0.9	\pm 8.63e-03	\pm 1.64e+00	\pm 2.98e-02	\pm 5.36e+02	\pm 2.39e+01	\pm 1.41e-02
50	0.9	0.1	\pm 0.00e+00	\pm 2.29e+00	\pm 0.00e+00	\pm 2.13e+01	\pm 0.00e+00	\pm 2.24e-15
30	0.5	0.9	\pm 1.59e+01	\pm 5.20e+01	\pm 3.42e+03	\pm 7.74e+02	\pm 5.09e+00	\pm 7.77e-01
30	0.5	0.1	\pm 0.00e+00	\pm 1.40e+01	\pm 0.00e+00	\pm 9.80e+01	\pm 0.00e+00	\pm 1.58e-30
10d	0.5	0.9	\pm 3.66e-04	\pm 4.75e-01	\pm 4.49e-04	\pm 2.80e+02	\pm 9.57e+00	\pm 2.06e-03
10d	0.5	0.5	\pm 9.09e-10	\pm 1.17e-01	\pm 8.00e-09	\pm 2.11e+02	\pm 8.09e+00	\pm 1.96e-06

Tablica 4.3. prikazuje rezultate pri različitim postavkama parametara i dimenzijom $d=30$, ovisno o funkcijama. Iz rezultata za sfernu funkciju može se uočiti da pet postavki parametara dolazi do optimalne vrijednosti. Isto tako iz tablice 4.4., gdje su prikazane standardne derivacije za svaki od rezultata, vidi se da svaki od tih postavki parametara koje su došle do optimalne vrijednosti ima standardnu devijaciju nula. To znači da algoritam za ove parametre i sfernu funkciju konzistentno dolazi do optimalne vrijednosti. Za Rosenbrockovu funkciju pokazano je kako niti jedna postavka parametara nije došla do optimalne vrijednosti u danoj vrijednosti $NFEs_{max}$. Najbolja postavka je $NP=100$, $F=0.5$, $CR=0.9$, što je jedna od najčešćih izbora za kombinaciju parametara, međutim ni ona nije došla do optimalnog rješenja. Kod Zakharove funkcije rezultati su slični rezultatima

Sferne funkcije. Iste postavke parametara konzistentno dolaze do globalnog optimuma funkcije kao i kod Sferne funkcije. Schwefelova funkcija je multimodalna te je teže pronaći globalni optimum funkcije. Međutim kombinacija parametara $NP=100$, $F=0.5$, $CR=0.3$ konzistentno dolazi do približne optimalne vrijednosti za ovaj problem. Rastriginova funkcija dolazi do optimalne vrijednosti za dvije postavke parametara, $NP=50$, $F=0.9$, $CR=0.1$ i $NP=30$, $F=0.5$, $CR=0.1$. Može se uočiti da obje postavke imaju vrlo malu vrijednost stope križanja. Kod Ackleyeve funkcije pet postavki parametara dolazi vrlo blizu optimalne vrijednosti. To su iste postavke parametara kao i kod Sferne funkcije. Iz ovih rezultata može se zaključiti da za većinu ovih problema kombinacija $NP=100$, $F=0.5$, $CR=0.9$ daje najbolje rezultate. U četiri od šest funkcija za ovu kombinaciju parametara dostiže se optimalna vrijednost. Isto tako, ostale četiri kombinacije koje su postigle optimalnu vrijednost kod Sferne funkcije dolaze do optimalnih ili blizu optimalnih vrijednosti i kod drugih funkcija.

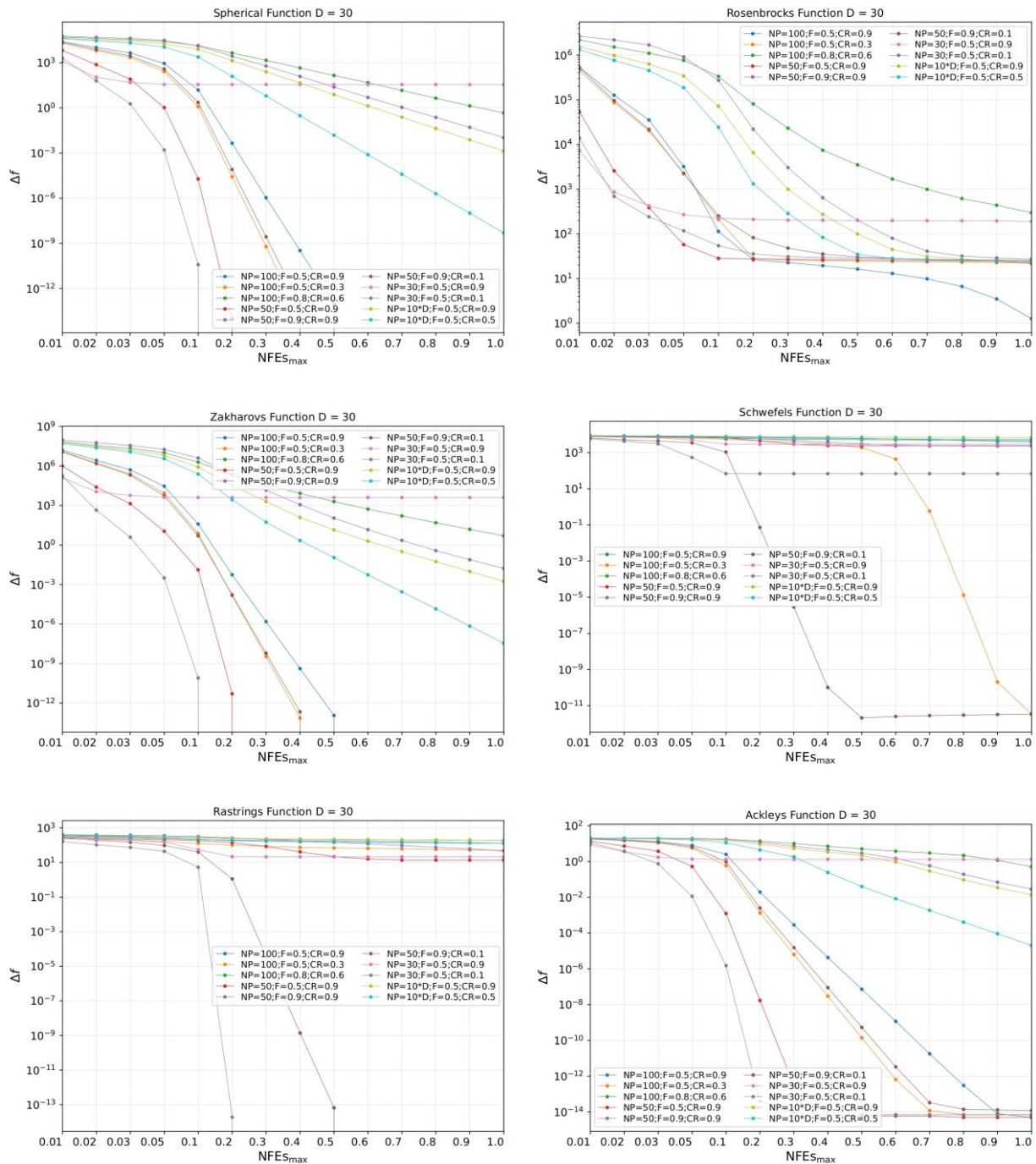
Tablica 4.5. Rezultati postavki parametara ovisno o funkcijama sa $d=50$

NP	F	CR	Sferna funkcija	Rosenbrockova funkcija	Zakharova funkcija	Schwefelova funkcija	Rastriginova funkcija	Ackleyeva funkcija
100	0.5	0.9	0.000e+00	2.170e+01	0.000e+00	7.335e+03	1.969e+02	7.000e-15
100	0.5	0.3	0.000e+00	4.066e+01	0.000e+00	7.100e+03	1.746e+02	9.667e-15
100	0.8	0.6	3.009e+03	1.361e+05	8.377e+05	1.039e+04	4.629e+02	1.104e+01
50	0.5	0.9	0.000e+00	5.991e+01	0.000e+00	5.084e+03	3.204e+01	1.564e-01
50	0.9	0.9	9.725e-03	5.659e+01	2.484e-01	5.349e+03	6.041e+01	1.933e-02
50	0.9	0.1	0.000e+00	4.484e+01	0.000e+00	2.731e+01	0.000e+00	2.397e-14
30	0.5	0.9	4.017e+01	3.789e+02	3.310e+05	5.574e+03	4.841e+01	3.987e+00
30	0.5	0.1	0.000e+00	4.905e+01	0.000e+00	2.243e+02	6.633e-02	1.380e-14
10d	0.5	0.9	6.771e+01	4.731e+02	1.758e+03	1.349e+04	4.163e+02	3.375e+00
10d	0.5	0.5	9.704e-02	1.551e+02	9.079e+00	1.155e+04	3.225e+02	9.885e-02

Tablica 4.6. Standardne devijacije rezultata postavki parametara ovisno o funkcijama sa $d=50$

NP	F	CR	Sferna funkcija	Rosenbrockova funkcija	Zakharova funkcija	Schwefelova funkcija	Rastriginova funkcija	Ackleyeva funkcija
100	0.5	0.9	$\pm 0.00e+00$	$\pm 9.96e+00$	$\pm 0.00e+00$	$\pm 1.42e+03$	$\pm 3.93e+01$	$\pm 1.58e-30$
100	0.5	0.3	$\pm 0.00e+00$	$\pm 5.87e-01$	$\pm 0.00e+00$	$\pm 4.42e+02$	$\pm 8.46e+00$	$\pm 1.89e-15$
100	0.8	0.6	$\pm 3.30e+02$	$\pm 2.63e+04$	$\pm 1.62e+05$	$\pm 3.01e+02$	$\pm 1.73e+01$	$\pm 4.42e-01$
50	0.5	0.9	$\pm 0.00e+00$	$\pm 2.95e+01$	$\pm 0.00e+00$	$\pm 9.07e+02$	$\pm 7.09e+00$	$\pm 3.51e-01$
50	0.9	0.9	$\pm 1.05e-02$	$\pm 2.00e+01$	$\pm 2.49e-01$	$\pm 9.55e+02$	$\pm 2.47e+01$	$\pm 8.72e-03$
50	0.9	0.1	$\pm 0.00e+00$	$\pm 1.60e+00$	$\pm 0.00e+00$	$\pm 4.66e+01$	$\pm 0.00e+00$	$\pm 2.82e-15$
30	0.5	0.9	$\pm 8.78e+01$	$\pm 7.51e+02$	$\pm 8.15e+05$	$\pm 7.69e+02$	$\pm 1.06e+01$	$\pm 1.13e+00$
30	0.5	0.1	$\pm 0.00e+00$	$\pm 1.64e+01$	$\pm 0.00e+00$	$\pm 2.12e+02$	$\pm 2.48e-01$	$\pm 7.48e-16$
10d	0.5	0.9	$\pm 1.34e+01$	$\pm 9.62e+01$	$\pm 6.31e+02$	$\pm 4.00e+02$	$\pm 1.58e+01$	$\pm 2.02e-01$
10d	0.5	0.5	$\pm 1.08e-02$	$\pm 1.72e+01$	$\pm 9.83e-01$	$\pm 3.12e+02$	$\pm 1.20e+01$	$\pm 7.97e-03$

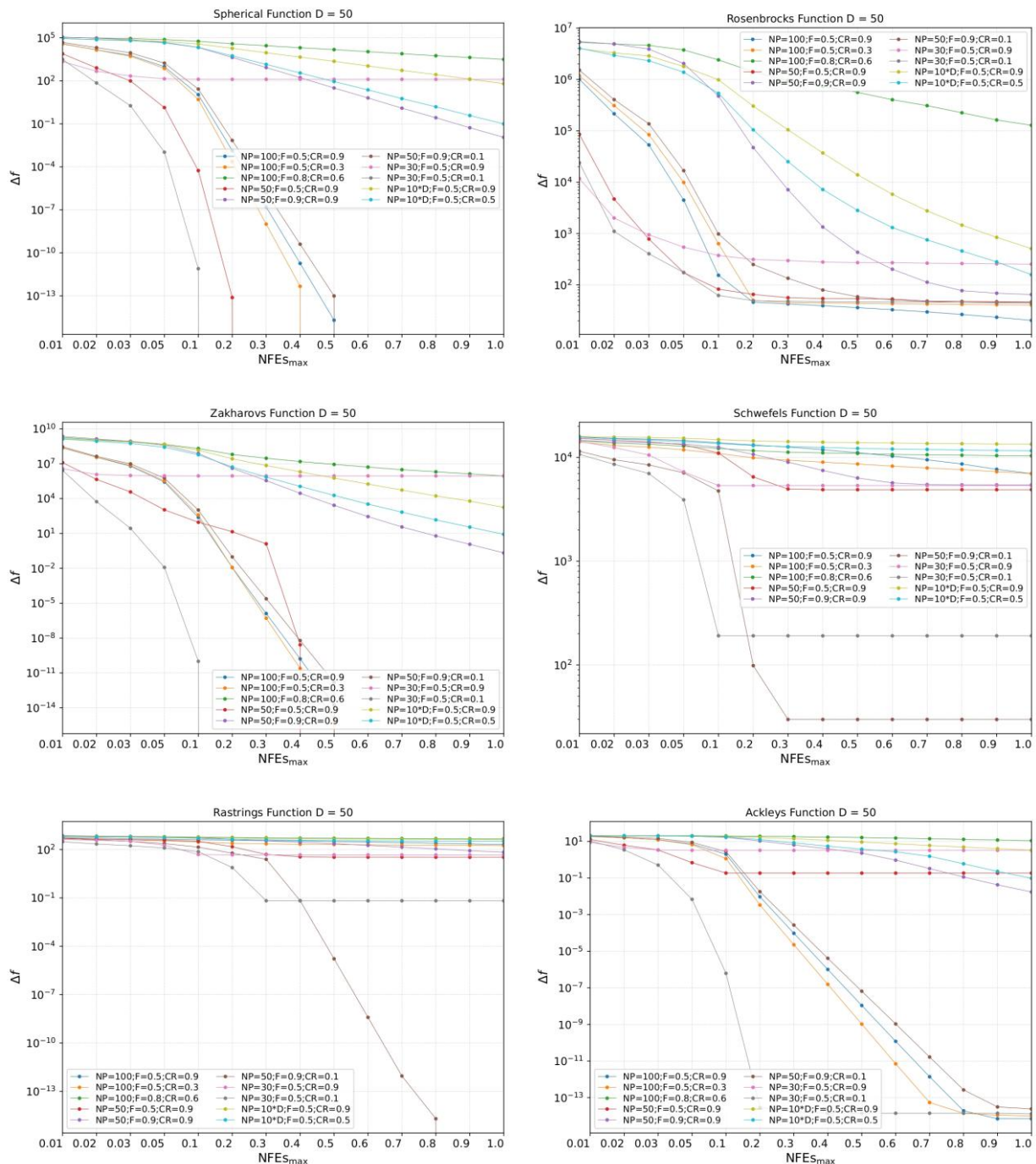
Tablice 4.5. i 4.6. prikazuju rezultate kada je dimenzija $d=50$, te njihove standardne devijacije. Za Sfernu funkciju rezultati su slični kao i kod $d=30$, pet kombinacija parametara dolazi do optimalne vrijednosti, dok su za ostale kombinacije rezultati lošiji. Kod Rosenbrockove funkcije ista kombinacija parametara dolazi do najbolje vrijednosti, ali isto tako niti jedna kombinacija ne dolazi do optimalne vrijednosti funkcije. Očekivano je da su rezultati lošiji zbog veće složenosti problema. Zbog toga, najbolji rezultat pri dimenziji 50 lošiji je nego pri dimenziji 30. Rezultati Zakharove funkcije su slični kao i u prethodnoj dimenziji. Razlika između dimenzija 50 i 30 uočava se kod Schwefelove funkcije. Najbolja kombinacija parametara je $NP=50, F=0.9, CR=0.1$, dok je pri manjoj dimenziji to bila kombinacija $NP=100, F=0.5, CR=0.3$. Međutim, najbolji rezultat pri manjoj dimenziji je daleko bolji, te dolazi blizu optimalne vrijednosti dok pri dimenziji 50, najbolji rezultat ne dolazi blizu optimalne vrijednosti. Rastriginova funkcija za kombinaciju parametara $NP=50, F=0.9, CR=0.1$ konzistentno dolazi do optimalne vrijednosti, ali za razliku od rezultata pri manjoj dimenziji, gdje je kombinacija $NP=30, F=0.5, CR=0.1$ došla do optimalne vrijednosti. U slučaju dimenzije 50, navedena kombinacija ne dolazi do optimalne vrijednosti. Za Ackleyevu funkciju rezultati su slični kao i za dimenziju 30. Povećanjem dimenzije rezultati kombinacija parametara koje nisu postigle optimalne vrijednosti su lošije nego pri manjoj dimenziji.



Sl. 4.1. Grafovi konvergencije pri $d=30$

Na slici 4.1. prikazani su grafovi svih korištenih funkcija cilja. Osim prikaza kombinacija parametara koje su došle do optimalne vrijednosti može se dobiti bolji uvid i napraviti usporedba različitih kombinacija ovisno o funkciji. Na taj način, kod sferne funkcije može se primijetiti da je pet najboljih kombinacija došlo do optimalne vrijednosti do 50% vrijednosti $NFEs_{max}$. Te se ističe kombinacija koja dovodi do stagnacije. Vrijednost funkcije cilja pada, ali nakon toga ostaje na istoj vrijednosti do kraja izvođenja algoritma. Kod Rosenbrockove funkcije vidi se pojava stagnacije na gotovo svim kombinacijama parametara. Zakharova funkcija je, kao što je bilo

vidljivo i u prethodnim tablicama, vrlo slična sfernoj funkciji. Kod Schwefelove funkcije dvije kombinacije parametara dobivaju dobre rezultate, ali ne dostiže se optimalna vrijednost već dolazi do stagnacije. Rastriginova funkcija dolaze do optimalnih vrijednosti za dvije kombinacije parametara do 50% $NFEs_{max}$. Ostale kombinacije stagniraju. Rezultati Ackleyeve funkcije pokazuju da za pet kombinacija parametara dobiva dobre rezultate, ali ne dolaze do optimalne vrijednosti, već stagniraju.



Sl. 4.2. Grafovi konvergencije pri $d=50$

Prema rezultatima na slici 4.1. može se zaključiti da one kombinacije koje ne dođu do optimalnih vrijednosti do $50\% NFE_{s_{max}}$ ne uspijevaju konvergirati niti do kraja izvođenja algoritma. Slika 4.2. prikazuje grafove konvergencije rezultata korištenih funkcija pri $d=50$. Rezultati na većoj dimenziji su slični rezultatima na $d=30$, iako postoji par razlika. Najbitnija razlika je rezultat Rastriginove funkcije. Jedna kombinacija koja je dostigla optimalnu vrijednost u ovom slučaju ju ne dostiže već dolazi do stagnacija. Ističe se od ostalih zato što ipak dolazi do manjih vrijednosti. Isto tako druga kombinacija parametara, $NP=50$, $F=0.9$, $CR=0.1$, dolazi do optimalne vrijednosti znatno kasnije. 4.2.

5. ZAKLJUČAK

Zadatak ovog završnog rada bio je istražiti ponašanje i učinkovitost algoritma diferencijalne evolucije u optimizaciji različitih funkcija, uz fokus na utjecaj parametara poput veličine populacije, faktora skaliranja i stope križanja. Cilj je bio analizirati kako postavke ovih parametara utječu na učinkovitost algoritma pri rješavanju unimodalnih i multimodalnih optimizacijskih problema pri različitim dimenzijama.

Tijekom rada ostvareno je programsko rješenje koje omogućuje korisnicima postavljanje parametara algoritma diferencijalne evolucije i funkcija cilja. Izvođenjem eksperimenta prikupljeni su podatci o kvaliteti rješenja. Podatci su dani korisniku u obliku tablice, te je dana mogućnost vizualizacija podataka s pomoću grafova. Eksperimentalna analiza pokazala je velik utjecaj postavki parametara algoritma diferencijalne evolucije na ponašanje samog algoritma. Ovisno o postavkama mijenjala se brzina konvergencije, došlo je do pojave stagnacije i promijene kvalitete rješenja. Neke postavke pokazale su se kao dobar izbor u većini funkcija. Time se pokazao utjecaj i važnost pravilnog odabira postavki parametara za problem koji se rješava. Odabirom krivih postavki dolazi do stagnacije rješenja. Dok odabir pravih postavki omogućuje optimizaciju problema unutar 50% vrijednosti $NFEs_{max}$.

Za budući rad preporučuje se proširenje analize na dodatne funkcije cilja, te ispitivanje više kombinacija parametara. Isto tako moguće je korištenje metoda koje podešavaju parametre algoritma tijekom samog izvođenja.

LITERATURA

- [1] A. P. Engelbrecht, Computational Intelligence: An Introduction, Second Edition. Wiley Publishing, 2007.
- [2] V.-H. Truong i S.-E. Kim, "Reliability-based design optimization of nonlinear inelastic trusses using improved differential evolution algorithm," *Advanced Engineering Software*, svez. 121, 2018, pp. 59-74.
- [3] N. H. Awad, M. Z. Ali, R. Mallipeddi i P. N. Suganthan, "An efficient differential evolution algorithm for stochastic opf based active–reactive power dispatch problem considering renewable generators," *Applied Soft Computing*, svez. 76, 2019, pp. 445-458.
- [4] G. Strofylas, K. Porfyri, I. Nikolos, A. Delis i M. Papageorgiou, "Using synchronous and asynchronous parallel differential evolution for calibrating a second-order traffic flow model," *Advanced Engineering Software*, svez. 125, 2018, pp. 1-18.
- [5] C. Lins, D. Eckhoff, A. Klausen, S. Hellmers, A. Hein i S. Fudickar, "Cardiopulmonary resuscitation quality parameters from motion capture data using differential evolution fitting of sinusoids," *Applied Soft Computing*, svez. 79, 2019, pp. 300-309.
- [6] R. Storn i K. Price, Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces," technical report, International Computer Science, Berkley, Kalifornija, 1995.
- [7] R. Storn i K. Price, Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces, *Journal of Global Optimization*, svez. 11, br. 4, pp. 341–359, 1997.
- [8] K. Price, R. Storn i J. A. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*, Springer-Verlag New York, Inc., 2005.
- [9] A. E. Eiben i J. E. Smith, *Introduction to Evolutionary Computing*, 2nd ed. ur., Springer Publishing Company, Incorporated, 2015.
- [10] D. Bajer, G. Martinović i J. Brest, "A population initialization method for evolutionary algorithms based on clustering and Cauchy deviates", *Expert Systems with Applications*, svez. 60, pp. 294–310, 2016.
- [11] D. Bajer, "Adaptive k-tournament mutation scheme for differential evolution", *Applied Soft Computing*, svez. 85, 2019.
- [12] A. Draa, K. Chettah i H. Talbi, "A compound sinusoidal differential evolution algorithm for continuous optimization," *Swarm and Evolutionary Computation*, svez. 50, pp. 100450, 2019.
- [13] S. Das, S. S. Mullick i P. N. Suganthan, "Recent advances in differential evolution – an updated survey", *Swarm and Evolutionary Computation*, svez. 27, pp. 1–30, 2016.
- [14] D. Bajer, "Parameter control for differential evolution by storage of successful values at an individual level", *Journal of Computational Science*, svez. 68, pp. 101985, 2023.
- [15] A. E. Eiben, R. Hinterding i Z. Michalewicz, "Parameter control in evolutionary algorithms", *IEEE Transactions on Evolutionary Computation*, svez. 3, br. 2, pp. 124–141, 1999.
- [16] C. Huang, Y. Li i X. Yao, "A survey of automatic parameter tuning methods for metaheuristics", *IEEE Transactions on Evolutionary Computation*, svez. 24, br. 2, pp. 201–216, 2020.

SAŽETAK

Ovaj rad bavi se istraživanjem diferencijalne evolucije, jednog od evolucijskih algoritama koji se pokazao vrlo učinkovit za probleme kontinuirane optimizacije. Rad uključuje analizu učinkovitosti algoritma diferencijalne evolucije pri različitim postavkama parametara. Parametri koje se postavljaju su veličina populacije, faktor skaliranja i stopa križanja. Detaljno su opisani parametri, te su navedene česte postavke istih. Opisuje se ostvareno programsko rješenje, te način rada samog rješenja. Eksperimentalna analiza provedena je na tri unimodalne i tri multimodalne funkcije, te su korištene dvije dimenzionalnosti u svrhu istraživanja ponašanja algoritma u rješavanju različitih problema. Analizirali su se rezultati, s posebnim fokusom na prosječne rezultate i standardne devijacije. Zaključno, rad pokazuje kako promjene u parametrima algoritma mogu utjecati na njegovu učinkovitost, te pruža smjernice o odabira postavki parametara ovisno o problemu.

Ključne riječi: diferencijalna evolucija, evolucijski algoritmi, kontinuirana optimizacija, parametri algoritma

ABSTRACT

Impact of the parameters of differential evolution on its efficiency

This thesis investigates differential evolution, one of the evolutionary algorithms that has proven highly effective for continuous optimization problems. The paper includes an analysis of the efficiency of the differential evolution algorithm under various parameter settings. The parameters considered are population size, scaling factor, and crossover rate. The parameters are described in detail, and common settings are provided. The implemented software solution is described, as well as its functionality. The experimental analysis was conducted on three unimodal and three multimodal functions, with two different dimensionalities, in order to explore the algorithm's behavior in solving various problems. The results were analyzed with a particular focus on average outcomes and standard deviations. In conclusion, the thesis demonstrates how changes in the algorithm's parameters can affect its efficiency, and provides guidelines for selecting parameter settings depending on the problem.

Keywords: differential evolution, evolutionary algorithms, continuous optimization, algorithm parameters

ŽIVOTOPIS

Kristijan Turić rođen je 2002. godine u Pakracu. Srednjoškolsko obrazovanje završio je u Kutini, nakon čega je upisao 1. godinu preddiplomskog studija Računarstva na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija. Tijekom studija pokazao je posebno zanimanje za područja poput računalnog programiranja i izrade softvera. Aktivno je sudjelovao u nekoliko projekata vezanih za razvoj web i mobilnih aplikacija. Pored akademskih projekata radio je na usavršavanju znanja u području izrade računalnih programa koristeći programske jezike kao što su C i C#.

Potpis autora