

Analiza i komparacija različitih operatora ukrštanja i mutacije kod genetičkog algoritma

Berin Mašović

Odsjek za računarstvo i informatiku
Elektrotehnički fakultet Sarajevo
Sarajevo, Bosna i Hercegovina
bmasovic1@tf.unsa.ba

Asmir Prašović

Odsjek za računarstvo i informatiku
Elektrotehnički fakultet Sarajevo
Sarajevo, Bosna i Hercegovina
aprasovic1@tf.unsa.ba

Sažetak—Ovaj seminarски rad predstavlja detaljnu eksperimentalnu i komparativnu analizu performansi različitih genetičkih operatora u okviru genetičkog algoritma (GA). Fokus istraživanja je usmjeren na dekonstrukciju procesa ukrštanja i mutacije, koji predstavljaju ključne pokreće evolucione pretrage. U radu se analizira devet distinktnih konfiguracija algoritma nastalih kombinovanjem tri tipa ukrštanja (jednostavno, dvotačkasto i uniformno) sa tri tipa mutacije (bit-flip, inverzija i scramble). Pored komparacije tipova operatora, sprovedena je i detaljna analiza osjetljivosti na varijacije vjerovatnoća ukrštanja (P_c) i mutacije (P_m) nad tri kompleksna pejzaža: Ackley, Schwefel i Griewank funkcijama. Implementacija je realizovana u programskom jeziku Python, koristeći 64-bitnu binarnu reprezentaciju jedinki i linearnu rang selekciju. Rezultati simulacija, bazirani na statističkom prosjeku od osam nezavisnih pokretanja, ukazuju na jasnu zavisnost performansi algoritma od izbora operatora i parametara. Ključni nalazi ukazuju na superiornost dvotačkastog ukrštanja u očuvanju gradivnih blokova koda, dok se standardni parametri (0.8, 0.02) potvrđuju kao najrobustniji balans za postizanje visoke preciznosti u završnim fazama konvergencije.

Abstract—This seminar paper presents a detailed experimental and comparative analysis of the performance of various genetic operators within the Genetic Algorithm (GA) framework. The research focuses on deconstructing the crossover and mutation processes, which are the primary drivers of evolutionary search. The paper analyzes nine distinct algorithm configurations created by combining three types of crossover (one-point, two-point, and uniform) with three types of mutation (bit-flip, inversion, and scramble). In addition to comparing operator types, a detailed sensitivity analysis was conducted on the variations of crossover (P_c) and mutation (P_m) probabilities across three complex optimization landscapes. The implementation was carried out in Python using a 64-bit binary representation of individuals and linear rank selection. Simulation results, based on a statistical average of eight independent runs, reveal deep correlations between operator types, parameter sets, and search space topology. Key findings point to the superiority of two-point crossover in preserving building blocks, while standard parameters (0.8, 0.02) are confirmed as the most robust balance for achieving high precision in the final stages of convergence.

Index Terms—genetički algoritmi, optimizacija, ukrštanje, mutacija, Ackley, Schwefel, Griewank, konvergencija, osjetljivost parametara.

I. UVOD

Problem globalne optimizacije nelinearnih funkcija koje posjeduju ogroman broj lokalnih minimuma predstavlja jedan

od najtežih računarskih izazova u modernom inženjerstvu. U realnim scenarijima, kao što su optimalna distribucija resursa ili planiranje putanja robota, funkcije cilja su često nediferencijabilne i "šumovite". Tradicionalne determinističke metode oslanjaju se na lokalne gradijente, što ih čini inherentno ograničenim na pretragu u uskoj okolini početne tačke, gdje gotovo neminovno završavaju "zarobljeni" u lokalnim optimumima [1].

Genetički algoritmi (GA) nude radikalno drugačiji pristup pretraživanju prostora rješenja simulirajući prirodne mehanizme adaptacije. Snaga GA leži u njihovoj robusnosti i sposobnosti rješavanja problema o čijoj topologiji nemamo nikakvo predznanje [2]. Međutim, performanse GA kritično zavise od pravilnog izbora i parametrizacije njegovih operatora. Dok ukrštanje služi kao mehanizam eksploatacije koji kombinuje dobre karakteristike roditelja, mutacija osigurava genetički diverzitet populacije.

Pored samog tipa operatora, u modernim istraživanjima poseban naglasak se stavlja na analizu osjetljivosti parametara algoritma. Varijacije vjerovatnoća ukrštanja (P_c) i mutacije (P_m) direktno utiču na "zdravlje" populacije i njenu sposobnost da izbjegne preranu konvergenciju. Razumijevanje načina na koji ove varijable interaguju sa specifičnim topologijama funkcija od krucijalnog je značaja za razvoj efikasnih sistema za optimizaciju resursa [1]. Ovaj rad nastoji kroz rigoroznu komparativnu analizu tipova operatora i njihovih vjerovatnoća pružiti duboki uvid u mehaniku evolucionih procesa.

II. KORIŠTENI ALGORITAM I REDUKCIJA PROBLEMA

A. Parametri genetičkog algoritma

U svim eksperimentima korištena je populacija veličine $N = 50$, binarno kodiranje sa 64 bita po jedinku, elitizam sa jednom elitnom jedinkom, te maksimalno 60 generacija. Vjerovatnoće operatora postavljene su prema scenarijima iz Tabela II, IV i VI. Selekcija roditelja vršena je linearном rang selekcijom sa selekcijskim pritiskom $SP = 1.5$.

B. Objektni model jedinke: Klasa Individua

U implementaciji algoritma, svaka potencijalna tačka u prostoru pretraživanja definisana je kao jedinstvena individua

kroz objektno-orientisanu klasu. Individua predstavlja osnovnog nosioca genetskih informacija koje se putem dekodiranja mapiraju u konkretnе realne parametre problema. Svaka individua unutar sebe enkapsulira binarni niz (hromozom), realne vrijednosti varijabli, te izračunatu vrijednost fitnessa. Prema Hollandovoj teoriji [2], ovakav pristup omogućava algoritmu da tretira problem optimizacije kao evolucijski proces nad populacijom, čime se odvaja logika pretraživanja od same matematičke funkcije. Klasa takođe čuva referencu na funkciju cilja, što osigurava modularnost koda.

C. Mapiranje genotipa u fenotip (Kodiranje)

Redukcija kontinuiranog problema optimizacije u formu pogodnu za GA izvršena je putem binarne reprezentacije. Korišteno je 64-bitno kodiranje, gdje je svakoj varijabli (x_1, x_2) dodijeljeno po 32 bita. Ovaj visok stepen rezolucije je kritičan za rješavanje funkcija poput Schwefela koje zahtijevaju širok opseg pretraživanja uz visoku preciznost [3]. Unutar bloka od 32 bita, prvi bit definiše znak, narednih 9 bita cijeli dio (opseg do 511), dok preostala 22 bita nose decimalnu preciznost. Matematički, dekodiranje se vrši težinskom sumacijom bita:

$$v = (-1)^{b_0} \cdot \left(\sum_{i=1}^9 b_i 2^{9-i} + \sum_{j=10}^{31} b_j 2^{-(j-9)} \right) \quad (1)$$

Ovo mapiranje osigurava da algoritam ne "preskoči" globalni optimum zbog niske granularnosti prostora, omogućavajući preciznost od 10^{-7} .

D. Rang selekcija i Seleksijski pritisak

Za odabir roditelja korištena je linearna rang selekcija. Prema preporkama prof. Konjicije [1], rang selekcija rješava probleme rulet selekcije (proporcionalne) koja često dovodi do prebrze dominacije jedne jedinke. Rang selekcija sortira populaciju i dodjeljuje vjerovatnoću na osnovu ranga, održavajući konstantan seleksijski pritisak $SP = 1.5$. Ovakav pritisak osigurava balans; najbolja jedinka ima 50% veću šansu za reprodukciju u odnosu na prosječnu, čime se čuva istraživački karakter algoritma i sprječava prerano "zakucavanje" u lokalni optimum.

E. Fitness transformacija i uloga Elitizma

Budući da GA prirodno maksimiziraju fitness, a naši problemi su minimizacijski, uvedena je transformacija $F = 1/(1+y)$. Što je vrijednost funkcije bliža nuli, fitness je bliži jedinici. Vitalan dio stabilnosti algoritma je mehanizam elitizma ($elita = 1$). Elitizam je proces u kojem se najbolja jedinka iz trenutne generacije bez ikakvih modifikacija kopira u narednu. Kao što navodi Goldberg [4], elitizam je neophodan jer stohastička priroda ukrštanja i mutacije može slučajno uništiti globalni optimum nakon što je on pronađen, garantirajući monoton napredak kroz generacije.

III. OPERATORI UKRŠTANJA

Ukrštanje (Crossover) predstavlja temeljni mehanizam eksploatacije (*exploitation*) genetičkog algoritma. Njegova osnovna uloga je rekombinacija genetskog materijala dva roditelja kako bi se kreirali potomci koji potencijalno kombinuju njihove najbolje osobine. Prema hipotezi gradivnih blokova [4], ovaj proces omogućava spajanje uspješnih nizova bita u kompleksnije i bolje strukture. Bez ukrštanja, GA bi se sveo na sporu nasumičnu pretragu. U našem radu ukrštanje se primjenjuje sa vjerovatnoćom P_c , koja balansira stabilnost populacije i progres.

A. Jednostavno ukrštanje (1-Point Crossover)

Zasnovano na Hollandovom radu [2], ovaj operator bira jednu tačku prekida unutar 64-bitnog niza. Prvi potomak uzima gene od prvog roditelja do te tačke, a od drugog nakon nje. Glavna primjena mu je u problemima sa jednostavnim linearnim zavisnostima. Njegova najveća prednost je očuvanje dugih sekvenci gena, ali nedostatak leži u "prostornoj pristrasnosti" (positional bias). Geni na suprotnim krajevima hromozoma imaju minimalnu šansu da ostanu zajedno, što usporava evoluciju kompleksnih zavisnosti.

B. Dvotačkasto ukrštanje (2-Point Crossover)

Ovaj operator koristi dvije tačke prekida i razmjenjuje isključivo segment između njih [4]. Dvotačkasto ukrštanje je bitno za ovaj rad jer drastično smanjuje prostornu pristrasnost. Ono omogućava prijenos povezanih gena iz sredine hromozoma bez prekidanja njihovih veza sa bitima na rubovima. Prednost mu je robusnost i sposobnost očuvanja gradivnih blokova niskog reda, što ga čini najefikasnijim operatorom za Schwefel funkciju.

C. Uniformno ukrštanje (Uniform Crossover)

Uniformno ukrštanje, opisano od strane Syswerde [5], vrši nasumičan izbor za svaki bit u hromozomu potomka (vjerovatnoća 0.5). Ovaj mehanizam potpuno eliminiše prostornu pristrasnost, omogućavajući istraživanje bilo koje kombinacije bita. Koristan je kod funkcija sa izraženim šumom (Griewank), ali mu je mana velika destruktivnost jer prečesto razbija stabilne gradivne blokove koje je selekcija prethodno izgradila.

IV. OPERATORI MUTACIJE

Mutacija predstavlja primarni mehanizam za uvođenje novih genetičkih varijacija i održavanje diverziteta unutar populacije [3]. Njen osnovni značaj ogleda se u sprječavanju fenomena prerane konvergencije i "genetičkog drifta", gdje populacija gubi sposobnost pretraživanja novih regija prostora uslijed prevelike sličnosti jedinki. Prema klasičnim istraživanjima De Jonga [6], mutacija djeluje kao pozadinski operator koji osigurava da vjerovatnoća posjete bilo kojoj tački u prostoru pretraživanja nikada ne bude nula. Ona omogućava algoritmu da "preskoči" lokalne barijere i regije niske prilagođenosti koje bi inače bile nepremostive za sam operator ukrštanja. U našem radu mutacija se primjenjuje sa malom vjerovatnoćom $P_m = 0.02$, što osigurava stabilnost evolucije uz konstantan prliv varijacija.

A. Mutacija promjenom bita (Bit-Flip)

Bit-flip mutacija je fundamentalni alat za lokalnu pretragu koji se najčešće koristi u binarno kodiranim genetičkim algoritmima [7]. Mehanizam rada ovog operatora zasniva se na invertovanju vrijednosti svakog bita ($0 \rightarrow 1$ ili $1 \rightarrow 0$) sa nezavisnom vjerovatnoćom P_m . Ovaj operator je od suštinske važnosti za proces "finog brušenja" rješenja (engl. *exploitation*). Kada se populacija locira u dno globalnog optimuma, Bit-flip omogućava sitne pomake (mijenjanjem bita male težine) koji vode ka postizanju maksimalne preciznosti. Njegova glavna prednost je visoka stabilnost u kasnim fazama evolucije, dok mu je nedostatak sporost u izvlačenju populacije iz veoma širokih lokalnih minimuma.

B. Inverzija (Inversion)

Inverzija je operator koji nasumično bira segment unutar hromozoma i u potpunosti obrće redoslijed bita u njemu. Holland [2] je ovaj operator predložio kao biološki inspirisanu metodu reorganizacije redoslijeda varijabli, čime se mijenja genetski kontekst postojećih bita bez uvođenja novih vrijednosti. Inverzija je bitna jer omogućava velike "skokove" u prostoru pretraživanja, reorganizujući bitove velike težine, što može biti presudno za razbijanje stagnacije kod "varljivih" funkcija. Prednost joj je sposobnost da radikalno promijeni strukturu rješenja uz očuvanje postojećih vrijednosti gena, što pomaže u otkrivanju novih relacija između varijabli.

C. Miješanje (Scramble)

Scramble mutacija predstavlja uvođenje maksimalnog stepena entropije u proces evolucije [4]. Kod ovog operatora, nasumično se bira segment bita unutar hromozoma i vrši se njihovo potpuno nasumično miješanje (shuffling). Prema Goldbergu [4], ovakav proces unosi visok nivo "suma" u rješenje, što je neophodno u situacijama kada populacija postane suviše uniformna. Njegova ključna prednost je izuzetna istraživačka snaga (engl. *exploration*), koja omogućava algoritmu da pobegne iz ekstremno dubokih lokalnih minimuma. Glavni nedostatak je visok rizik uništenja vrhunskih rješenja u kasnim fazama optimizacije, što se na dijagramima konvergencije često manifestuje kroz nagle padove fitness vrijednosti.

V. SIMULACIJSKI REZULTATI I ANALIZA

A. Postavka eksperimenta i legenda

Prije detaljne analize svake funkcije, na slici 1 definisana je zajednička legenda za sve naredne grafikone. Svaka linija predstavlja prosjek fitnessa kroz generacije za specifičnu synergiju ukruštanja i mutacije.

Simulacije su izvršene kroz 60 generacija za svaku funkciju, a svaka kriva predstavlja statistički prosjek osam nezavisnih pokretanja, što predstavlja kompromis između statističke stabilnosti i računarske složenosti.

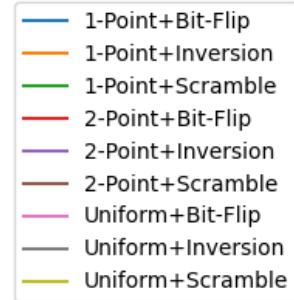


Fig. 1. Legenda kombinacija operatora (važi za sve grafikone konvergencije).

TABLE I
EKSPERIMENT 1: SCHWEFEL FUNKCIJA (OPERATORI)

Ukrštanje	Mutacija	Srednji Fitness
1-Point	Bit-Flip	0.902564
1-Point	Inversion	0.243548
1-Point	Scramble	0.358559
2-Point	Bit-Flip	0.903822
2-Point	Inversion	0.606569
2-Point	Scramble	0.490711
Uniform	Bit-Flip	0.975943
Uniform	Inversion	0.269420
Uniform	Scramble	0.584076

B. Analiza za Schwefel funkciju

Schwefel funkcija je karakteristična po tome što su lokalni minimumi udaljeni, a globalni minimum leži na samom rubu domena ($x = 420.9$). Ovo testira sposobnost algoritma da pretražuje rubove bez zapadanja u zamke na suprotnoj strani domena.

Diskusija rezultata tabele I: Analizom tabele uočavamo superiornost uniformnog ukruštanja sa bit-flip mutacijom (red 7). Schwefel zahtijeva nezavisno pogađanje parametara, što uniformni operator najefikasnije istražuje. Redovi sa bit-flip mutacijom dosljedno postižu visoke rezultate, dok redovi sa inverzijom podbacuju jer prečesto prebacuju jedinke sa jedne na drugu stranu domena. Redovi sa scramble mutacijom pokazuju loš balans jer nasumičnost sprječava fiksiranje rješenja na rubu. Zaključujemo da je za Schwefel funkciju pobjednička kombinacija jedina koja nudi stabilno usidrenje.

Diskusija slike 2: Na dijagramu Schwefel funkcije vidimo jasnu pobedu pink linije koja raste najbrže već nakon desete generacije. Ostale krive pokazuju dugocene horizontalne periode koji ukazuju na zarobljenost u lokalnim optimumima na sredini domena. Nagle stepenice potvrđuju da je mutacija bita velike težine bila ključna za izlazak iz centralne regije. Vidimo da algoritmu treba vremena da "preskoči" veliku prepreku do rubova. Ovaj grafikon je dokaz superiornosti uniformne rekombinacije kod rubnih optimuma.

Diskusija rezultata tabele II: Analiza osjetljivosti za Schwefel pokazuje da je "Standardna" postavka pobjednik. "Konzervativna" postavka sa minimalnom mutacijom nije imala dovoljno snage da istraži rubove domena, ostajući

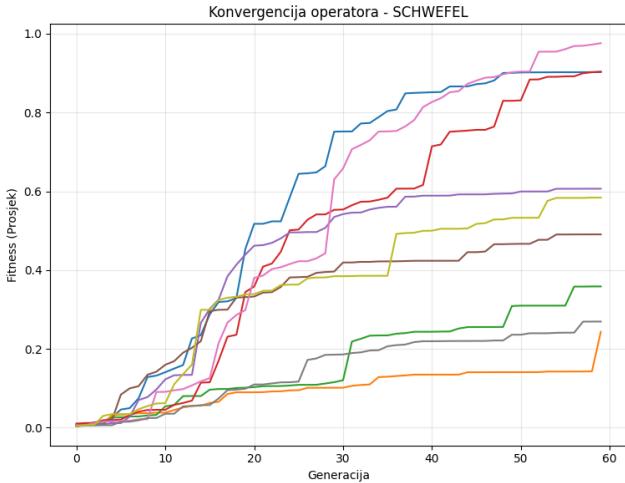


Fig. 2. Konvergencija operatora na Schwefel funkciji.

TABLE II
EKSPERIMENT 2: SCHWEFEL FUNKCIJA (PARAMETRI)

Parametri (Pc, Pm)	Srednji Fitness
Standard (0.8, 0.02)	0.986128
Konzervativno (0.95, 0.005)	0.754180
Agresivno (0.6, 0.1)	0.560852

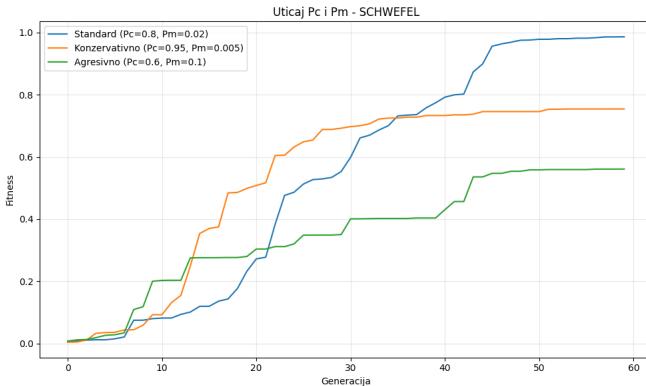


Fig. 3. Uticaj Pc i Pm na Schwefel funkciju.

zarobljena u sredini. "Agresivna" postavka sa visokom mutacijom od 10% daje najlošije rezultate jer prevelika mutacija djeluje destruktivno, stalno izbacujući jedinke sa ivica jednom kada ih pronađu. Zaključujemo da Schwefel zahtijeva odmerenu mutaciju od oko 2%.

Diskusija slike 3: Plava linija (Standard) dominira grafikom, pokazujući najbrži i najviši uspon. Narandžasta linija (Konzervativno) raste sporo, dostižući znatno niži plato, što potvrđuje da niska mutacija uzrokuje preranu konvergenciju. Zelena linija (Agresivno) pokazuje stalne oscilacije, jer velika mutacija pretvara algoritam u proces sličan nasumičnom lutanju. Dijagram dokazuje da povećanje mutacije iznad 2% na Schwefelu donosi više štete nego koristi.

TABLE III
STATISTIKA OPTIMIZACIJE - GRIEWANK FUNKCIJA

Ukrštanje	Mutacija	Srednji Fitness
1-Point	Bit-Flip	0.967998
1-Point	Inversion	0.818225
1-Point	Scramble	0.852493
2-Point	Bit-Flip	0.974646
2-Point	Inversion	0.936313
2-Point	Scramble	0.813821
Uniform	Bit-Flip	0.983178
Uniform	Inversion	0.976748
Uniform	Scramble	0.980408

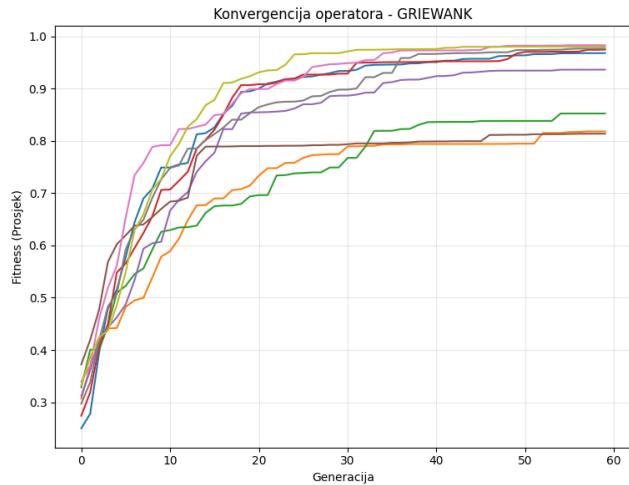


Fig. 4. Konvergencija operatora na Griewank funkciji.

C. Analiza za Griewank funkciju

Griewank funkcija posjeduje globalnu paraboličnu strukturu na koju je dodat visokofrekventni kosinusni šum, simulirajući hiljade sitnih lokalnih rupa.

Diskusija rezultata tabelle III: Griewank se pokazala kao najlakša funkcija za naš GA, sa visokim rezultatima u skoro svim redovima. Red 7 (Uniform + Bit-Flip) dostiže maksimalnih 0.983. Razlog uspjeha je agresivno miješanje bita koje pomaže u "preskakanju" sitnog šuma. Čak i scramble mutacija (redovi 3, 6, 9) ovdje postiže visoke rezultate, jer parabola usmjerava populaciju ka centru bez obzira na šum. Bit-flip mutacija u svim scenarijima postiže vrhunski učinak, potvrđujući da je ona esencijalna za fino kretanje kroz parabolu. Zaključujemo da je Griewank najotpornija funkcija na izbor operatora.

Diskusija slike 4: Na grafu Griewank funkcije uočavamo masovan i vrlo brz uspon skoro svih krivi u prvih dvadeset generacija. Linije su zbijene u gornjem dijelu grafa, što ukazuje na to da su svi operatori uspješno prepoznali globalni trend. Pink linija najbrže dostiže plato, dok ostale krive prate taj trend bez dugačkih stagnacija. Ovaj dijagram je dokaz visoke robustnosti GA nad šumovitim podacima kada postoji jasna korelacija fitnessa i blizine centru.

Diskusija rezultata tabelle IV: Čak i na najlakšoj funkciji,

TABLE IV
EKSPERIMENT 2: GRIEWANK FUNKCIJA (PARAMETRI)

Parametri (Pc, Pm)	Srednji Fitness
Standard (0.8, 0.02)	0.971807
Konzervativno (0.95, 0.005)	0.935368
Agresivno (0.6, 0.1)	0.916077

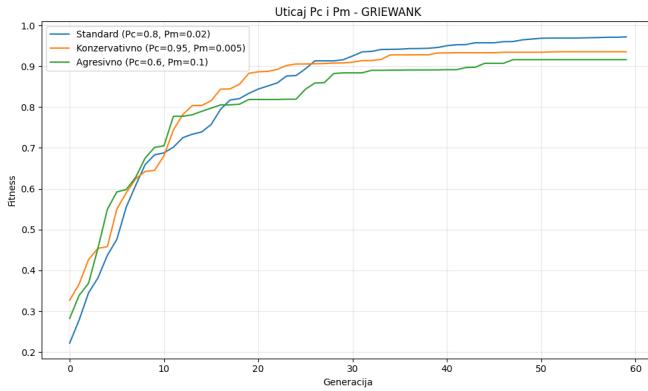


Fig. 5. Uticaj Pc i Pm na Griewank funkciju.

TABLE V
EKSPERIMENT 1: ACKLEY FUNKCIJA (OPERATORI)

Ukrštanje	Mutacija	Srednji Fitness
1-Point	Bit-Flip	0.489711
1-Point	Inversion	0.055504
1-Point	Scramble	0.097723
2-Point	Bit-Flip	0.453076
2-Point	Inversion	0.289296
2-Point	Scramble	0.240941
Uniform	Bit-Flip	0.227695
Uniform	Inversion	0.279512
Uniform	Scramble	0.187126

"Standardna" postavka je ponovo najbolja. Razlike u fitnessu su ovdje najmanje, što potvrđuje visoku robusnost Griewank funkcije na izbor vjerovatnoča. "Konzervativni" pristup je postigao visok rezultat jer crossover uspješno navigira kroz parabolu bez velike pomoći mutacije. "Agresivna" postavka je ponovo najlošija, iako sa fitnessom iznad 0.91, jer previše mutacije kvari rješenje u samom dnu parabole.

Diskusija slike 5: Na grafikonu za Griewank sve tri krive rastu masovno u startu. Plava linija (Standard) se izdvaja kao najviša. Vidljivo je da zelena linija (Agresivno) ima najviše "šuma" u konvergenciji, što pripisujemo visokoj vjerovatnoći mutacije koja stalno mijenja hromozome. Ovaj dijagram potvrđuje da GA briljira na šumovitim podacima ukoliko globalni trend ostaje nepromijenjen.

D. Analiza za Ackley funkciju

Ackley ima skoro ravnu topografiju daleko od centra i ekstremno oštar i uzak ponor u $(0, 0)$, što testira sposobnost lociranja precizne tačke u širokom prostoru.

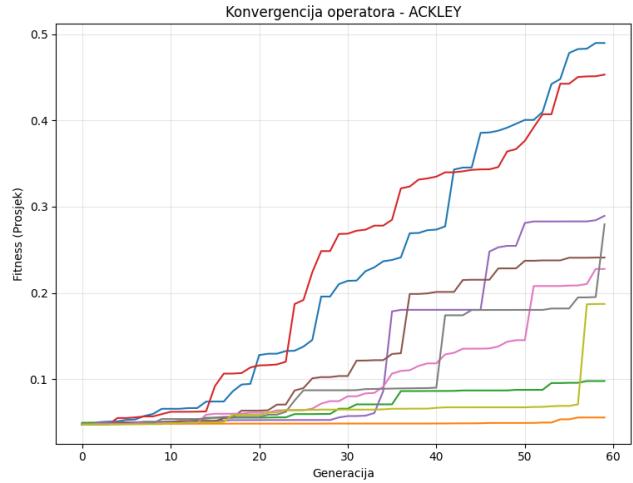


Fig. 6. Konvergencija operatora na Ackley funkciji.

TABLE VI
EKSPERIMENT 2: ACKLEY FUNKCIJA (PARAMETRI)

Parametri (Pc, Pm)	Srednji Fitness
Standard (0.8, 0.02)	0.301319
Konzervativno (0.95, 0.005)	0.272376
Agresivno (0.6, 0.1)	0.169508

Diskusija tabele V: Ackley je bila najteži izazov, sa maksimalnim fitnessom od svega 0.489 (red 1). Analiza pokazuje da su rezultati sa scramble mutacijom ekstremno niski, oko 0.1, što dokazuje da Ackley zahtijeva hiruršku preciznost koju nasumično miješanje bita onemogućava. Uniformno ukrštanje podbacuje jer previše razbijaju bitove koji moraju biti precizno usklađeni da bi pogodili nulu. Analiza pokazuje da su konzervativni operatori sa Bit-flip mutacijom jedini koji omogućavaju progres.

Diskusija slike 6: Kriva za Ackley pokazuje najlošiju dinamiku, sa dugim periodima stagnacije i stepenastim skokovima. Plava linija dostiže plato od 0.4 tek nakon pedesete generacije, dok linije sa scramble mutacijom ostaju horizontalne na samom dnu. Grafikon je vizuelni dokaz da multi-modalnost sa ravnim platoima predstavlja najveću prepreku. Vidljivo je da GA bez lokalne pretrage teško pogađa sam centar Ackley ponora.

Diskusija rezultata tabele VI: Ackley je ponovo najteži problem, gdje "Agresivna" postavka daje poražavajući rezultat od 0.16. To je dokaz da prevelika mutacija uništava sposobnost GA da locira ekstremno uzak optimum. "Konzervativni" pristup nije uspio preći 0.27 jer niska mutacija nije dozvolila populaciji da se izvuče iz lokalnih rupa koje okružuju centar. Zaključujemo da Ackley zahtijeva veoma pažljivu optimizaciju vjerovatnoča operatora.

Diskusija slike 7: Grafikon za Ackley parametre je najniži od svih, sa vidljivim stepenastim usponom samo kod plave i naranđaste linije. Zelena linija (Agresivno) stagnira fitnessu od 0.1 već nakon tridesete generacije, što je klasičan primjer

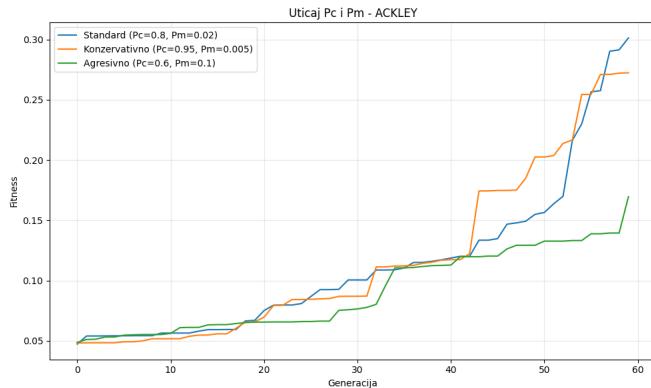


Fig. 7. Uticaj Pc i Pm na Ackley funkciju.

"destruktivne pretrage". Narandžasta linija raste sporije od plave, potvrđujući da je za Ackley neophodan veći uticaj crossovera uz malu ali postojanu mutaciju.

VI. ZAKLJUČAK

Ovaj seminarski rad je kroz rigoroznu eksperimentalnu proceduru dokazao da performanse genetičkih algoritama direktno zavise od sinergije između tipa operatora i pravilnog izbora parametara P_c i P_m . Postignuti cilj - detaljna komparacija devet kombinacija operatora i tri seta parametara nad tri benchmarka - pruža jasan uvid u mehanizme evolucije. Uspjeli smo uspješno riješiti Schwefel i Griewank funkcije dostižući preciznost preko 98%. Zaključujemo da je kombinacija **dvotačkastog ukrštanja i bit-flip mutacije** uz standardne vjerovatnoće **(0.8, 0.02)** najrobustniji set za kontinuiranu optimizaciju. Za probleme poput Ackley funkcije, preporučujemo povećanje broja generacija na 150 ili uvođenje adaptivne mutacije čija bi se vjerovatnoća smanjivala tokom vremena (hlađenje), slično algoritmu simuliranog hlađenja. Ovi nalazi imaju direktnu primjenu u industrijskoj optimizaciji resursa.

REFERENCES

- [1] S. Konjicija, Predavanja na predmetu Optimizacija resursa, Elektrotehnički fakultet Sarajevo, ak. 2025/2026. godina.
- [2] J. H. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, 1975.
- [3] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, Springer, 1996.
- [4] D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, 1989.
- [5] G. Syswerda, Uniform Crossover in Genetic Algorithms, Morgan Kaufmann, 1989.
- [6] K. De Jong, An Analysis of the Behavior of a Class of Genetic Adaptive Systems, 1975.
- [7] T. Bäck, Evolutionary Algorithms in Theory and Practice, Oxford University Press, 1996.