

Univerzitet u Beogradu
Matematički fakultet

Računarska inteligencija
Seminarski rad

**Meta-optimizacija genetskog
algoritma za rekonstrukciju slike**

Sažetak

Rad se bavi optimizacijom parametara genetskog algoritma koji prosledenu sliku rekonstruiše nizom obojenih figura zadatog tipa. Rekonstrukcija slike se dobija iscrtavanjem niza figura. Fitnes jedne rekonstrukcije izračunava se njenim poređenjem sa originalnom slikom. Meta-optimizacija je izvedena genetskim algoritmom koji pronalazi kombinaciju vrednosti parametara poboljšavanjem kvaliteta rekonstrukcije. Fitnes jedinke se izračunava sprovođenjem rekonstrukcije sa vrednostima parametara zadatih jedinkom.

Studentkinja: Tijana Živković 66/2018

Sadržaj

1	Uvod	2
1.1	Opis problema	2
1.2	Genetski algoritmi. Meta-optimizacija.	2
2	Rešenje problema	4
2.1	Genetski algoritam za rekonstrukciju slike	7
2.2	Genetski algoritam za meta-optimizaciju	8
3	Kvalitet rešenja	9
4	Zaključak	10

1 Uvod

1.1 Opis problema

Osnovni problem kojim se ovaj rad bavi je rekonstrukcija rasterske slike zadate u .jpg, .jpeg ili .png formatu nizom jednoboynih dvodimenzionih figura istog tipa pozicioniranim u granicama zadatih dimenzijama originalne slike. Figure mogu biti proizvoljnih veličina koje zavise jedino od dimenzija polazne slike, tako da nijedna figura ne izlazi iz zadatog okvira. Broj figura je unapred zadat. Cilj je odrediti boju, veličinu i poziciju svake u nizu figura tako da slika koja se dobija njihovim iscrtavanjem predstavlja dovoljno dobru rekonstrukciju polazne slike.

Ovako postavljen problem se može posmatrati kao problem optimizacije. Prostor pretrage je skup svih mogućih rasporeda zadatog broja figura svih mogućih boja i veličina. Dopustivo rešenje je niz figura sa pozicijama i veličinama takvim da ne izlaze iz okvira zadatog dimenzijama originalne slike i bojama koje se mogu reprezentovati npr. RGB kolor modelom. Od svih dopustivih rešenja cilj je naći takav niz figura čijim se iscrtavanjem dobija slika što sličnija originalnoj slici, pa je prema tome funkcija cilja sličnost (ili različitost) sa originalnom slikom koju algoritam optimizacije maksimizuje (ili minimizuje).

1.2 Genetski algoritmi. Meta-optimizacija.

Evolutivna izračunavanja predstavljaju paradigmu računarske inteligencije i teorijsku osnovu imaju u Darvinovoj teoriji biološke evolucije. Biološku evoluciju simuliraju implementacijom koncepata prirodne selekcije, reprodukcije, mutacije gena i drugih, određenim operatorima i njihovom iterativnom primenom nad populacijom kandidata rešenja datog optimizacionog problema. Evolutivni algoritam je **populaciona** (održava populaciju kandidata rešenja) **metaheuristika** (opšta strategija pretrage za rešavanje optimizacionih problema formulisana nezavisno od konkretnog problema [2]) koja vrši **stohastičku pretragu** (slučajnost figuriše u pretrazi) prostora svih mogućih rešenja.[1][3]

Inicijalne ideje o primeni znanja o biološkoj evoluciji u optimizaciji su se pojavile još 1950ih. Ključni doprinos razvoju genetskih algoritama dao je Džon Holand (John Holland) koji je 1970ih godina razvio kanonski genetski algoritam. Pored Holanda, razvoju evolutivnih izračunavanja doprineli su i Fogel (Lawrence Fogel) razvojem evolutivnog programiranja i Rekenberg

(Rechenberg) razvojem evolutivnih strategija (evolucija evolucije).[1][3]

Osnovni aspekti evolutivnog algoritma[1][3]:

- Jedna jedinka u populaciji predstavlja jedno moguće rešenje problema. Hromozom predstavlja karakteristike (gena) jedne jedinke. Rešenje problema se kodira u vidu hromozoma, tako da svaki gen odgovara jednoj karakteristici rešenja. Vrednost jedne karakteristike, odnosno sadržaj jednog gena naziva se genski alel. Genotip opisuje vrednosti karakteristika jedne jedinke, tj. koje alele hromozom poseduje.
- Koncept prilagođenosti jedinke implementiran je fitnes funkcijom koja evaluira kvalitet pojedinačnog rešenja. Fitnes funkcija je matematička funkcija koja hromozomu dodeljuje skalarnu vrednost. Prema Darvinovoj teoriji, bolje prilagođene jedinke imaju veću šansu da prežive dovoljno dugo i ostave potomstvo. Prema tome, fitnes funkcija ima pozitivan smisao - što je veći fitnes hromozoma, to je njime reprezentovano rešenje bolje. Fitnes funkcija je obično, ali ne i nužno, jednaka funkciji cilja.
- Inicijalizacija početne populacije jedinki je najčešće nasumična, odnosno čini je skup nasumično odabranih rešenja iz skupa dopustivih rešenja.
- Operator selekcije definiše način na koji se biraju jedinke koje ulaze u proces reprodukcije, koja podrazumeva ukrštanje i, ne nužno, mutaciju.
- Operator ukrštanja definiše način na koji se vrši rekombinacija roditeljskih gena u cilju stvaranja novih jedinki.
- Evolutivni algoritam se sprovodi iterativno, sve dok nije ispunjen unapred zadat kriterijum zaustavljanja. Jedna iteracija podrazumeva formiranje novih jedinki sprovođenjem selekcije i reprodukcije nad trenutnom populacijom. Novu populaciju čine jedinke odabrane primenom određene politike zamene generacija.

Genetski algoritmi su jedna od paradigmi evolutivnih izračunavanja koja podrazumeva reprezentaciju rešenja u vidu niza karakteristika (linearan genotip). Genetski algoritmi imaju primenu u rešavanju problema iz različitih domena i predstavljaju dobru strategiju za rešavanje problema za koja ne postoje efikasna egzaktna rešenja. Glavnu primenu imaju u rešavanju problema u diskretnom domenu. Kao i većina populacionih metaheuristika, genetski algoritmi mogu da budu vremenski i prostorno veoma zahtevni. Genetski

algoritam ne garantuje pronalaženje globalno optimalnog, pa čak ni lokalno optimalnog rešenja, ali je uglavnom rešenje koje pronađe dovoljno dobro.[3][2]

Genetski algoritam se instancira za konkretan problem definisanjem, najpre, reprezentacije rešenja konkretnog problema, a zatim i svih ostalih aspekata (fitnes funkcije, operatora ukrštanja, operatora selekcije, itd.), kao i odabirom neophodnih parametara. Za neke reprezentacije jedinki postoje već razvijeni operatori koji se mogu koristiti bez modifikacije. Brzina pretrage i kvalitet rešenja koje algoritam pronalazi u velikoj meri zavise od odabranih parametara.[2]

Parametri genetskog algoritma se mogu ručno podešavati, što je vrlo nepraktično. Problem pronalaženja adekvatnih vrednosti parametara genetskog algoritma se može formulisati kao problem optimizacije i samim tim parametri se mogu optimizovati nekom tehnikom optimizacije.[2] S obzirom na to da je genetski algoritam metaheuristika koja se koristi kao optimizaciona tehnika, optimizacija parametara genetskog algoritma je primer tzv. meta-optimizacije. Meta-optimizacija se može izvesti i nekom metaheuristikom.[4]

Meta-optimizacija obuhvata dva nivoa: meta-nivo i bazni nivo. Na meta-nivou optimizuju se parametri optimizacione tehnike na baznom nivou. Jedno rešenje na meta-nivou predstavlja vrednosti parametara, a funkcija cilja na meta-nivou se formuliše u zavisnosti od funkcije cilja na baznom nivou.[4] Na primer, meta-optimizacija genetskog algoritma se može sprovesti genetskim algoritmom. Jedan hromozom genetskog algoritma na meta-nivou kodira jedan niz vrednosti parametara genetskog algoritma na baznom nivou. Fitnes jedinke na meta-nivou izračunava se kao fitnes najbolje jedinke (najboljeg rešenja) koju pronalazi genetski algoritam na baznom nivou sa vrednostima parametara zadatih jedinkom na meta-nivou.

2 Rešenje problema

Za rekonstrukciju slike razvijen je genetski algoritam čiji hromozomi kodiraju rekonstrukcije originalne slike. Jedan hromozom predstavlja niz figura čijim se iscrtavanjem dobija jedna rekonstrukcija. Unapred zadat broj figura definiše dužinu hromozoma. Broj figura je jedan od meta-parametara genetskog algoritma za rekonstrukciju slike. Fitnes jedne rekonstrukcije izračunava se njenim poređenjem sa originalnom slikom. Meta-optimizacija genetskog algoritma za rekonstrukciju slike izvedena je genetskim algoritmom. Jedan hromozom je niz meta-parametara potrebnih za rekonstruisanje slike. Fitnes

se izračunava sprovođenjem rekonstrukcije sa vrednostima parametara zadatih hromozomom. Algoritmi su implementirani u Python-u.

U cilju pojednostavljenja problema, uvedene su sledeće pretpostavke:

- Figura može biti kvadrat, krug ili trougao.
- Slike i njihove rekonstrukcije su u RGB modu.
- Figure mogu biti obojene bilo kojom bojom koju je moguće reprezentovati RGBA kolor modelom, s tim da ne smeju biti u potpunosti prozirne.
- Unutrašnjost i ivica figure su obojene istom bojom.

Definisana je hijerarhija klasa Figure, Square, Circle i Triangle, čije instance predstavljaju gene hromozoma genetskog algoritma za rekonstrukciju slike. Klase su napravljene tako da sadrže sve neophodne informacije za iscrtavanje figure, prateći funkcije za iscrtavanje i rad sa slikama u korišćenoj Pillow (PIL) biblioteci. Osim toga, klase sadrže metode za mutaciju boje, mutaciju veličine i mutaciju pozicije figure, kao i metod kojim se sprovede sve navedene mutacije. Mutacije su implementirane na sledeći način:

- Mutacija boje podrazumeva dodavanje nasumično odabrane vrednosti iz fiksiranog intervala za svaki od R, G i B kanala. Slično, vrednost za alfa kanal mutira dodavanjem nasumično odabrane vrednosti iz fiksiranog intervala.
- Mutacija veličine figure je za kvadrat i krug definisana kao dodavanje nasumično odabrane vrednosti iz fiksiranog intervala, s obzirom na to da je njihova veličina zadata dužinom stranice, odnosno prečnika. Trougao je zadat pozicijama temena, pa je mutacija veličine definisana kao uniformno skaliranje sa nasumično odabranim faktorom iz fiksiranog intervala. Skaliranje se sprovodi kao složena transformacija: najpre se centar trougla translira u koordinatni početak, zatim se izvrši skaliranje i na kraju se trougao translira u početni položaj.
- Mutacija pozicije podrazumeva translaciju za nasumično odabran vektor.
- Intervali iz kojih se nasumično biraju vrednosti definisani su tako da obezbeđuju malu promenu i obuhvataju i negativne vrednosti.
- Mutacijom se dobija dopustivo rešenje, a nedopustivo se odbacuje.

Genetski algoritam za rekonstrukciju slike i genetski algoritam za njegovu meta-optimizaciju prate Algoritam 1 i razvijeni su tako da važe:

- Inicijalizacija jedinki je nasumična.
- Selekcija se sprovodi tako da prvi roditelj ne može učestvovati u izboru za drugog roditelja.
- Ukrštanjem se od dva odabrana roditelja dobijaju dva potomka, koja zamenjuju svoje roditelje u narednoj populaciji.
- Oba potomka mutiraju sa malom verovatnoćom. Mutacija se sprovodi po svakom genu sa zadatom verovatnoćom mutacije.

Algoritam 1 Pseudokod implementiranih genetskih algoritama

if *koristi elitizam* **and** *veličina elitističke populacije neparna* **then**

 Uvećaj veličinu elitističke populacije za 1;

end if

Kreiraj i inicijalizuj populaciju jedinki;

Izračunaj fitnes svake jedinke u populaciji;

Kreiraj i inicijalizuj listu najboljih jedinki na praznu listu;

while *nije dostignut zadat broj generacija* **do**

if *koristi elitizam* **then**

 Sortiraj jedinke u populaciji opadajuće prema fitnesu;

 Kreiraj elitističku populaciju od najboljih jedinki;

 Kopiraj elitne jedinke u novu populaciju;

end if

while *nije popunjena nova populacija* **do**

 Izvrši selekciju ne uključujući elitističku populaciju;

 Izvrši ukrštanje odabranih roditelja;

 Izvrši mutaciju potomaka;

 Evaluiraj potomke i dodaj ih u novu populaciju;

end while

 Postavi populaciju na novu populaciju;

 Dodaj najbolju jedinku u populaciji u listu najboljih jedinki;

end while

Vrati najbolju jedinku iz liste najboljih jedinki kao rezultat;

2.1 Genetski algoritam za rekonstrukciju slike

Ulazni parametri algoritma su: originalna slika, tip figure, broj figura, broj generacija, veličina populacije, informacija o tome da li se koristi elitizam i veličina elitističke populacije, verovatnoća mutacije, metod ukrštanja i odgovarajući parametar neophodan za sprovođenje ukrštanja i metod selekcije i odgovarajući selekциони parametar.

Jedinka je predstavljena nizom zadatog broja figura istog zadatog tipa koje se inicijalizuju na nasumičnim pozicijama i veličinama tako da ne izlaze iz okvira zadatog dimenzijama originalne slike koja se rekonstruiše. Boja i transparentnost se nasumično određuju, tako da figura ne sme biti potpuno transparentna.

Fitness jedinke se izračunava kao indeks strukturalne sličnosti između originalne slike i rekonstrukcije koja se dobija iscrtavanjem niza figura koji predstavlja jedinku. Strukturalna sličnost, odnosno SSIM (structural similarity index measure) metrika se dobija kombinovanjem rezultata poređenja osvetljenosti, kontrasta i strukture dve slike.[5] Korišćena je implementacija SSIM metrike iz scikit-image biblioteke koja kao rezultat vraća srednju vrednost strukturalne sličnosti MSSIM (mean structural similarity index over the image). Funkcija koja izračunava MSSIM pozvana je sa vrednostima parametara korišćenim u [5].

Razvijena su dva metoda selekcije i oba obezbeđuju da se selekcija sprovodi tako da prvi roditelj ne učestvuje u izboru drugog roditelja. Prvi metod je implementacija turnirske selekcije koja podrazumeva da se najpre od svih jedinki u populaciji koje mogu učestvovati u selekciji nasumično odabere nekoliko, u zavisnosti od veličine turnira, a zatim od tako odabranih jedinki odabere ona sa najvećim fitnessom. Drugi metod je implementacija ruletske selekcije koja bira svaku jedinku sa verovatnoćom direktno proporcionalnom njenom fitnessu. Jedinkama se pre izbora dodeljuju verovatnoće odabira izračunate kao količnik fitnessa jedinke i sume fitness vrednosti svih jedinki koje mogu učestvovati u izboru.[1][3]

Algoritam primenjuje jedan od tri metoda ukrštanja. Implementirano je k -poziciono ukrštanje tako da, nakon nasumičnog odabira k pozicija, potomci nasleđuju alternirajuće delove roditeljskih nizova figura, prethodno podeljenih na $k + 1$ delova. Drugi metod ukrštanja je implementacija uniformnog ukrštanja. Za svaki gen, sa verovatnoćom 0.5 prvo dete dobija alel prvog roditelja, a drugo dete alel drugog roditelja, u suprotnom je obrnuto.[1][3]

Treći metod ukrštanja je mešovito ukrštanje razvijeno po uzoru na jednostavno aritmetičko ukrštanje [3]. Najpre se nasumično odabere pozicija p . Alele prvog roditelja do odabrane pozicije p nasleđuje prvi potomak, a alele drugog roditelja drugi potomak. Oba potomka na pozicijama nakon p imaju iste alele dobijene kombinovanjem roditeljskih alela na pozicijama nakon p . Kombinovanje se vrši u zavisnosti od parametra $\alpha \in (0, 1)$ prema formuli $\alpha * a_1 + (1 - \alpha) * a_2$, gde je a_1 alel prvog roditelja, a a_2 alel drugog. Formula se primenjuje za svaku od R, G, B, A vrednosti boje, poziciju i veličinu figure.

Mutacija se sprovodi po svakom genu hromozoma pojedinačno sa zadatom verovatnoćom mutacije i podrazumeva mutaciju boje, veličine i pozicije instance figure koja predstavlja jedan gen.

2.2 Genetski algoritam za meta-optimizaciju

Ulazni parametri algoritma su: originalna slika, tip figure, broj generacija, veličina populacije, informacija o tome da li se koristi elitizam i veličina elitističke populacije, verovatnoća mutacije, veličina turnira (selekcija je turnirska), domeni vrednosti koji se koriste u inicijalizaciji meta-parametara genetskog algoritma za rekonstrukciju slike (domeni za broj figura, broj generacija, veličinu populacije, metode ukrštanja, metode selekcije i maksimalna vrednost verovatnoće mutacije).

Jedinka je predstavljena nizom meta-parametara algoritma za rekonstrukciju slike. Inicijalizacija vrednosti je nasumična: broj figura, broj generacija i veličina populacije se inicijalizuju na nasumično odabrane vrednosti iz odgovarajućih fiksiranih intervala; nasumično se odabere turnirska ili ruletska selekcija; nasumično se odabere jedna od metoda ukrštanja - k -poziciono, uniformno ili mešovito ukrštanje; nasumično se odredi da li treba primeniti elitizam. Ukoliko se primenjuje elitizam, veličina elitističke populacije je nasumično odabrana vrednost između 5% i 25% veličine populacije, u suprotnom jednaka je nuli. Verovatnoća mutacije se nasumično odabere iz intervala $[0.01, \textit{zadata maksimalna vrednost}]$. U slučaju da je odabrana turnirska selekcija, veličina turnira se inicijalizuje na nasumičnu vrednost između 10% i 50% veličine populacije. Parametar k za k -poziciono ukrštanje se inicijalizuje nasumično između 1 i 30% veličine populacije, a parametar mešovitog ukrštanja α nasumično iz $(0, 1)$.

Fitnes jedinke se izračunava kao fitnes rezultata izvršavanja genetskog algoritma za rekonstrukciju slike pozvanog sa vrednostima parametara kodi-

ranih jedinkom. Selekcija je turnirska i identična je turnirskoj selekciji implementiranoj za genetski algoritam za rekonstrukciju slike. Ukrštanje je uniformno, i od implementacije koju koristi genetski algoritam za rekonstrukciju slike razlikuje se jedino u proveru dopustivosti rešenja. Provera je neophodna zbog toga što su neki meta-parametri koji su kodirani hromozomom međusobno zavisni. Nedopustiva rešenja se odbacuju.

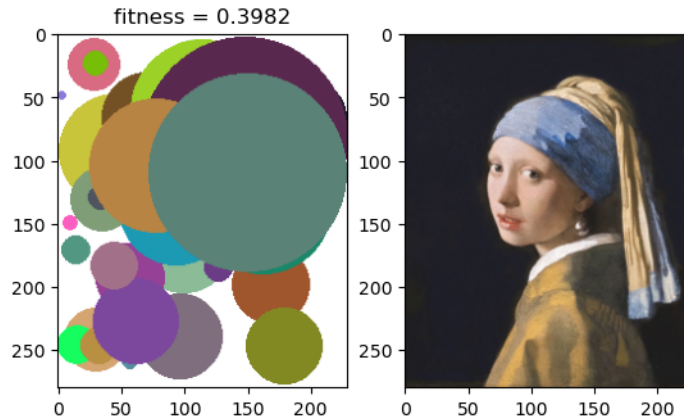
Mutacija se sprovodi po svakom genu hromozoma pojedinačno sa zadatom verovatnoćom mutacije i podrazumeva malu promenu vrednosti svakog od meta-parametara, a metode selekcije i ukrštanja mutiraju nasumičnim odabirom neke od preostalih metoda. Mutacijom se dobijaju isključivo dopustiva rešenja.

3 Kvalitet rešenja

Procena kvaliteta rada meta-optimizacionog genetskog algoritma izvršena je poređenjem njegovog rezultata sa rezultatom dobijenim genetskim algoritmom sa nasumično inicijalizovanim parametrima. Poređenje fitnesa rezultata sprovedeno je za svaku kombinaciju mogućih slika u test skupu i mogućih figura. Test skup sadrži 5 slika relativno malih dimenzija. Interval mogućih vrednosti za broj figura postavljen je na $[50, 200]$; broj generacija uzima neku vrednost iz $\{50, 60, \dots, 190\}$; veličina populacije je broj iz $\{10, 12, \dots, 198\}$; verovatnoća mutacije se bira iz intervala $[0.01, 0.2]$; selekcija je ili turnirska ili ruletska i ukrštanje je k -poziciono ili uniformno ili mešovito (opisano u 2.1). Veličina populacije meta-optimizacionog algoritma jednaka je 4, broj generacija 4, elitizam se primenjuje sa veličinom 2, a verovatnoća mutacije i veličina turnira jednake su redom 0.1 i 2. Meta-optimizacioni algoritam dao je bolje rezultate za 14 od 15 test instanci. Za test instancu gde su rezultati gori, fitnes rešenja je za 0.0047 manji od fitnesa rešenja koje pronalazi genetski algoritam za rekonstrukciju slike pozvan sa nasumično inicijalizovanim parametrima.

Meta-optimizacioni algoritam je, pre procene kvaliteta, testiran na nasumično odabranoj slici iz test skupa i nasumično odabranim tipom figure sa identičnim vrednostima parametara, osim za veličinu populacije koja je postavljena na 6. Algoritam uspeva da vrlo malo poboljša fitnes najbolje jedinke u populaciji kroz generacije, ili bar ne pogorša vrednost fitnesa (zbog primene elitizma), ali se dobijena rekonstrukcija i dalje u velikoj meri razlikuje od originalne slike (Slika 1). Genetski algoritam za rekonstrukciju slike je, posebno, testiran ručnim podešavanjem parametara i, takođe, rezultat koji

vraća ima neznatno veći fitness od najbolje jedinke u inicijalnoj populaciji i pronađena rekonstrukcija se znatno razlikuje od originalne slike.



Slika 1: Rezultat rada meta-optimizacionog algoritma

Za programiranje je korišćena Džupajter sveska (Jupyter Notebook), a sav rad izvršen je na virtualnoj mašini koja na raspolaganju ima 4.096GB RAM memorije i 6 procesorskih jezgara. Razvijen meta-optimizacioni algoritam je izuzetno vremenski zahtevan i zbog toga nisu izvršeni eksperimenti sa većim opsegom veličine populacije, broja generacija i figura.

4 Zaključak

Inicijalni pokušaj rekonstrukcije slike podrazumevao je neprozirne figure i izračunavanje fitnessa jedinke kao MSE (mean-squared error) vrednosti između originalne slike i rekonstrukcije predstavljene jedinkom i to pomnožene sa (-1), kako bi veći fitness označavao bolje prilagođenu jedinku. Iako su i algoritam za rekonstrukciju i meta-optimizacioni algoritam uspešno poboljšavali fitness vrednost najbolje jedinke, rekonstrukcije koje su vraćali kao rezultat prilično su odstupale od originalnih slika na kojima su testirani. U cilju poboljšanja rezultata, uvedena je transparentnost figura i odabrana je SSIM metrika za izračunavanje fitnessa jedinki. Međutim, ovo nije dovelo do poboljšanja.

Treba uzeti u obzir da nisu izvršeni eksperimenti sa većim brojem generacija, figura i veličine populacije. Povećanje opsega ovih parametara moglo bi da dovede do poboljšanja rezultata. Takođe, efikasnost algoritma bi se mogla

poboljšati uvođenjem paralelizacije, koja ovde nije implementirana. Dodatno, za bolju procenu rada, trebalo bi testirati algoritam na boljem sistemu.

Još jedna opcija za poboljšanje rekonstrukcije slike je redefinisane funkcije. Trebalo bi isprobati više razvijenih metoda za procenu sličnosti/različitosti slika ili možda kombinovati više metoda.

Definisanje boljeg metoda ukrštanja, umesto uniformnog ukrštanja koje meta-optimizacioni genetski algoritam primenjuje, moglo bi da pomogne u bržem pronalaženju bolje rekonstrukcije. Na primer, može se isprobati drugačiji redosled meta-parametara u hromozomu tako da međusobno zavisni parametri budu na bliskim pozicijama u nizu i tako iskoristiti pozicionu pristranost k -pozicionog ukrštanja. Po ugledu na [6] (uspešno rekonstruisanje slika genetskim algoritmom) moguće je izmeniti reprezentaciju jedinke i umesto niza figura održavati sliku u nizovskoj reprezentaciji i prema tome redefinisati ukrštanje i mutaciju. Ukrštanje je u navedenom primeru izvršeno direktno nad slikama i to razmenom niza piksela, a mutacija se sprovodi dodavanjem (iscrtavanjem) nasumične figure određenog tipa. Na tako izmenjenom genetskom algoritmu za rekonstrukciju slike bi meta-optimizacioni algoritam mogao biti testiran, uz potrebne izmene reprezentacije meta-parametara koje treba da optimizuje. Na taj način bi se možda mogla dobiti bolja procena kvaliteta rada meta-optimizacionog algoritma.

Literatura

- [1] A.P. Engelbrecht. *Computational Intelligence - An Introduction*, 2007.
- [2] P. Janičić i M. Nikolić. *Veštačka inteligencija*, 2024.
- [3] A. Kartelj. Slajdovi *Računarska inteligencija*, 2019.
- [4] E.S. Skakov and V.N. Malyshev. Journal of Physics: Conference Series. *Parameter meta-optimization of metaheuristics of solving specific NP-hard facility location problem*, <https://iopscience.iop.org/article/10.1088/1742-6596/973/1/012063/meta>, 2018.
- [5] Z. Wang, A.C. Bovik, H.R. Sheikh and E.P. Simoncelli. IEEE Transactions on Image Processing, 13, 600-612. *Image quality assessment: From error visibility to structural similarity.*, <https://ece.uwaterloo.ca/~z70wang/publications/ssim.pdf>, 2004.
- [6] N. Pachala. *Genetic algorithm image reconstruction*, <https://github.com/Th3NiKo/Genetic-algorithm-image-reconstruction>, 2020.