

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 3205

**Detekcija malignih tvorevina na
RTG slikama pluća uz pomoć
genetskog algoritma i
determinističkih algoritama
kontinuirane optimizacije**

Sofia Čolaković

Zagreb, srpanj 2013.

SADRŽAJ

1. Uvod	1
2. Implementacija rješenja	2
2.1. Općenito o genetskom algoritmu	2
2.2. Genetski algoritam s lokalnom pretragom	3
2.2.1. Prikaz optimiziranih parametara	3
2.2.2. Algoritmi lokalne pretrage	4
2.2.3. Operatori selekcije, križanja i mutacije GA	5
2.3. Hibridni algoritam	6
2.3.1. Operatori selekcije, križanja i mutacije	6
2.3.2. Genotip	8
2.4. Funkcija dobrote	8
2.5. Opis korištenih filtera	9
2.6. Korištena baza podataka	12
2.7. MATLAB	13
3. Rezultati	14
3.1. Genetski algoritam s lokalnom pretragom	14
3.1.1. Utjecaj veličine populacije	14
3.1.2. Nakon koliko generacija GA pokrenuti lokalnu prtragu	15
3.1.3. Uvjet zaustavljanja	16
3.1.4. Algoritmi lokalne pretrage	16
3.1.5. Utjecaj genotipa	17
3.2. Hibridni algoritam	17
3.2.1. Utjecaj veličine populacije	17
3.3. Ispitivanje najboljeg rješenja	18
4. Zaključak	20

1. Uvod

Dijagnoza malignih tvorevina na plućima kompliciran je postupak koji se vrši brojnim invazivnim i neinvazivnim metodama. Prije pristupanja biopsiji tkiva ili računalnoj tomografiji (CT) potrebno je već locirati tvorevinu na plućima koja upućuje na rak. Rendgensko snimanje pluća jeftina je i brza metoda koja ozračuje pacijenta malom dozom zračenja. To je prva dijagnostička metoda kojom se postavlja sumnja na rak pluća, te odlučuje da li pacijenta uputiti na daljnju dijagnostiku. Upravo zbog toga veoma je važno precizno detektirati čvoriće na RTG snimkama pluća, te odrediti da li su oni benigni ili maligni. Osjetljivost radiologa na plućne čvoriće na RTG snimkama pluća iznosi oko 70% uz osjetljivost od 85% na svojstvo detektiranog čvorića [1].

S obzirom na relativnu nepreciznost ljudske detekcije čvorića i važnost detekcije u ranoj fazi, postoji potreba za računalnim sustavom za obradu RTG slika i detekciju čvorića na plućima. Prirodom inspirirani algoritmi pogodni su općenito za obradu slike, ali i specifično za ovaj problem kada nemamo specifičan uzorak tvorevina niti algoritam za određivanje da li je čvorić benigni ili maligni.

Ovaj rad razmatra preciznost detekcije plućnih čvorića primjenom genetskog algoritma uz determinističke algoritme optimizacije.

2. Implementacija rješenja

Prilikom obrade, nad RTG slikom primjenjujemo filtere sa različitim parametrima. Nakon primjene filtera, na sliku primjenjujemo filter praga (*eng. threshold filter*). Filter praga pretvara sliku u crno-bijelu tako da piksele svjetline veće od praga boji bijelo, a ostatak crno. Tada bijeli pikseli predstavljaju područje čvorića, dok je crno područje bez čvorića. Genetski algoritam optimizira redoslijed filtera, koji će se filteri primijeniti te njihove parametre.

U ovom radu implementirana su tri različita rješenja za optimizaciju navedenih parametara. Dva genetska algoritma s različitim genotipima i lokalnom pretragom nakon svakih nekoliko generacija i hibridni algoritam. U ovom poglavlju izložit će se detalji implementacije tih algoritama.

2.1. Općenito o genetskom algoritmu

Genetski algoritam jest heuristička metoda optimizacije kojom se imitira evolucijski proces iz prirode. Na početku algoritma stvara se početna populacija, slučajno ili prema nekoj razdiobi. Sa svakom iteracijom algoritma izabiremo najpogodnije jedinke iz populacije, križamo ih, mutiramo rješenja, te djecu uzimamo kao populaciju sljedeće generacije. Svaka jedinka ima svoj genotip, koji jest rješenje problema i može se evaluirati, u ovom slučaju redoslijed i parametri filtera. Za svaku jedinku određujemo dobrotu prema kojoj je uspoređujemo s drugim jedinkama populacije. Za tu svrhu imamo funkciju dobrote koja svaku jedinku evaluira i dodjeljuje joj iznos njene dobrote.

Kako bi odredili genetski algoritam moramo odrediti njegove funkcije selekcije, križanja i mutacije. Funkcija selekcije određuje koje i koliko će algoritam jedniki iz populacije izdvojiti kao roditelje sljedeće generacije. Funkcija križanja odlučuje na koji ćemo način iz dvije roditeljske jedinke dobiti jedinku djeteta. Funkcija mutacije određuje hoćemo li mutirati jedinku i na koji ćemo način modificirati njen genotip. Mutacija omogućava raznovrsnost populacije, te sprječava prebrzo konvergiranje cijele

populacije u lokalnim minimumima.

2.2. Genetski algoritam s lokalnom pretragom

Genetski algoritam s lokalnom pretragom modifikacija je klasičnog genetskog algoritma kod koje se algoritam nakon fiksnog broja generacija zaustavlja i pokreće se algoritam lokalne pretrage. To se ponavlja s tim da nova iteracija genetskog algoritma uzima za početnu populaciju onu s kojom je završila prethodna iteracija algoritma. Algoritam lokalne pretrage može optimirati iste parametre kao i genetski algoritam, ali i ne mora.

U ovom radu genotip jedinki sastoji se od redoslijeda primjene filtera i cjelobrojnih parametara filtera, dok su se realni parametri filtera optimizirali lokalnom pretragom. U ovom slučaju genetski algoritam i algoritam lokalne pretrage optimizirali su dva odvojena skupa parametara. Deterministički algoritmi korišteni u lokalnoj pretrazi su simpleks algoritam po Nedleru i Meadu, te Hooke-Jeeves algoritam.

2.2.1. Prikaz optimiziranih parametara

Kako bi optimirali rješenje problema parametri koje optimiziramo moraju biti organizirani u primjerene strukture koje se mogu obrađivati navedenim algoritmom. Potrebno je prikazati redoslijed filtera koji se primjenjuju, te njihove parametre. Također, filteri imaju cjelobrojne i realne parametre, zbog čega ih nećemo moći optimirati istim algoritmom. Simpleks algoritam ne može optimirati cjelobrojne parametre, pa ćemo redoslijed filtera i cjelobrojne parametre optimirati s GA, a realne parametre determinističkim algoritmima. Parametri koje se optimiraju determinističkim algoritmom organizirani su kao vektor realnih brojeva čiji su elementi iznosi parametara, a svaki parametar ima određenu poziciju u vektoru. Genotipove genetskog algoritma definiramo na sljedeća dva načina.

Prikaz genotipa slijedom cijelih brojeva

$$q = [f_1 \ f_2 \dots f_n \ p_1 \dots p_m]$$

$f_1 - f_n$ - cjelobrojni id-jevi filtera u rasponu 1 - broj filtera

$p_1 - p_m$ - cjelobrojni parametri filtera

Kako bi se neki filter primjenio, njegov indeks mora se naći u dijelu vektora koji određuju redoslijed filtera. Filteri se primjenjuju slijedno kako se nalaze u vektoru. Filter čiji se indeks nalazi na prvom mjestu vektora primjenjuje se prvi. Ovakav način prikaza omogućava višestruko korištenje jednog filtera, u tom se slučaju oba puta on primjenjuje sa istim parametrima. Broj filtera koji će se primijeniti određuje se veličinom parametra n .

Prikaz genotipa slijedom realnih brojeva

$$q = [f_1 \ f_2 \dots f_{(\text{broj filtera})} \ t \ p_1 \dots p_m]$$

$f_1 - f_{(\text{broj filtera})}$ - parametri filtera u rasponu 0 - 1

t - prag kojim se odlučuje da li će se filter primijeniti ili ne, u rasponu 0 - 0.5

$p_1 - p_m$ - cjelobrojni parametri filtera

Prvi dio vektora čine realni brojevi u rasponu 0-1. Svaki broj predstavlja jedan filter. t je prag, te filteri čiji je realni parametar veći od praga se primjenjuju, dok se ostali ne primjenjuju. Filteri se primjenjuju redoslijedom određenim njihovim realnim parametrom, oni s većim parametrom primjenjuju se prije.

2.2.2. Algoritmi lokalne pretrage

Implementirana su dva algoritma lokalne pretrage, simpleks algoritam po Nedleru i Meadu i Hooke-Jeeves algoritam. Treba usporediti uspješnost ovih algoritama u optimizaciji opisanog problema s dva načina definiranja genotipa. Rezultati će dati najbolji način za obradu RTG slika, ali i naznaku u kojem bi se smjeru daljnja evolucija sustava za detekciju tumora mogla odvijati.

Simpleks algoritam po Nedleru i Meadu

Simpleks algoritam Nedlera i Meada jest algoritam heurističke optimizacije za višeparametrsku optimizaciju. Temelji se na konstrukciji simpleksa, konveksne ljuske oko zadane početne točke, početnog centroida. Konveksna ljuska ima n točaka koje su potencijalna rješenja problema. Iz točaka se svakom iteracijom izabire rješenje s najmanjom dobrotom, te se modificira funkcijama refleksije, ekspanzije i kontrakcije.

Kod funkcije refleksije, najslabije rješenje reflektiramo u odnosu na centroid. Ekspanzija je tome slična, osim što točku prebacujemo na daljinu veću nego kod refleksije.

Kontrakcijom smanjujemo udaljenost točke koju modificiramo i centroida. Algoritam ponavlja te korake dok ne dođe do konvergencije rješenja, što znači da je rješenje u lokalnom ili globalnom minimumu.

Hooke-Jeeves algoritam

Hooke-Jeeves algoritam je metoda optimizacije koja se temelji na pretraživanju okoline rješenja kako bi se našao minimum funkcije. U svakoj iteraciji algoritma istražuje se okolina trenutnog rješenja. Ako je nova pronađena točka bolja od stare, tj. bazne točke, bazno rješenje se prebacuje preko novoga. Ako je bazno rješenje bolje, ono se zadržava, te se smanjuje korak.

Funkcija koja istražuje okolinu rješenja radi tako da provjerava dobrotu parametra uvećanog za korak, te ako je ona lošija od početne, smanjuje vrijednost parametra na početnu umanjenu za korak. Ako je i ona lošija od početne točke, vraća vrijednost parametra na staro. Svaki od parametara koji se optimiziraju se tako istražuje. Smanjivanjem koraka istraživanja namještamo vrijednost parametra sve preciznije kako algoritam napreduje. Algoritam se zastaustavlja kada korak postane manji od neke predefinirane vrijednosti. Kao ulazni parametar algoritma uzima se ρ , faktor namještanja koraka u rasponu od 0-1. Korak se dobija kao umnožak ρ i koraka u prethodnoj iteraciji.

2.2.3. Operatori selekcije, križanja i mutacije GA

Za operatore genetskog algoritma uzeti su pretpostavljeni operatori Matlab ga funkcije za probleme koji sadrže realne parametre i ograničenja. Operatori GA su stohastička uniformna selekcija, prilagodljiva ostvariva mutacija i za funkciju križanja sritmetička sredina.

Stohastička uniformna selekcija

Definira se linija na kojoj svaki dio te linije predstavlja neku od jedinki populacije. Svaka jedinka ima dio linije čija je dužina proporcionalna njenoj dobroti, bolje jedinke imaju veći udio u ukupnoj duljini linije. Nakon toga definira se korak s obzirom na to koliko se jedinki bira iz populacije, te se po liniji kreće s definiranim korakom birajući pritom za roditelje sljedeće generacije jedinke na čijem je dijelu linije funkcija stala.

Funkcija križanja - aritmetička sredina

Funkcija križanja koja gen djeteta definira kao aritmetičku sredinu gena roditelja. Ova funkcija križanja pogodna je za probleme s realnim parametrima i ograničenjima.

Prilagodljiva ostvariva mutacija

Funkcija mutacije koja mijenja smjer mutacije s obzirom na zadnju generaciju. Oda- bire se smjer i iznos mutacije koji ne krše ograničenja pojedinih gena. Smjer se odabire s obzirom na to da li je zadnja generacija bila uspješna u pronalaženju boljeg rješenja ili nije.

2.3. Hibridni algoritam

Hibridni algoritmi su optimizacijski algoritmi koji imaju elemente prirodom inspirira- nih i determinističkih algoritama. U ovom radu implementiran je hibridni algoritam koji je kombinacija Hooke-Jeeves i genetskog algoritma. Radi se o algoritmu koji Hooke-Jeeves algoritmom optimizira jedinke populacije. Kada neka od jedinki ko- nvergira zamjenjuje se potomkom dviju jedinki iz populacije dobivenih operatorom selekcije. Također, nad novonastalom jedinkom primjenjuje se operacija mutacije.

2.3.1. Operatori selekcije, križanja i mutacije

Operator selekcije služi nam za odabir jedinke koju ćemo križati. Valja izabrati je- dinku što boljih karakteristika i ujedno osigurati raznovrsnost genotipa u populaciju. Operator selekcije odabire jedinku slučajno iz skupa trećine najboljih jedinki.

Operator križanja iz svakog para parametara genotipa roditelja stvara novi parame- tar. Interpolacija se provodi prema formuli: $x(i) = a(i) + rand * (b(i) - a(i))$

za svaki i parametar iz genotipa. x je genotip djeteta, dok su a i b genotipovi roditelja. $rand$ je slučajni parametar uniformne razdiobe koji određuje doprinos sva- kog od roditelja tom parametru genotipa i određuje se posebno za interpolaciju svakog parametra. Ako je parametar cjelobrojan on se određuje zaokruživanjem rezultata do- bivenog izrazom na najbliži cijeli broj.

Operator mutacije primjenjuje se nad parametrom genoma sa vjerojatnošću p_m koju zadajemo kao ulazni parametar algoritma. Ako se parametar mutira njegova se vrijednost zamjenjuje slučajno generiranom vrijednošću. Ako je parametar cjelobroj- ni nova vrijednost generira se po Poissonovoj, a za realni po uniformnoj razdiobi.

Algorithm 1 Hibridni algoritam

```
function HIBRIDNIALGORITAM( $sizePop, \varepsilon, \rho, p_m$ )  
   $P \leftarrow stvoriPopulaciju(sizePop)$   
   $P^{bazni} \leftarrow P$   
   $\Delta x \leftarrow [\rho * P_1, \dots, \rho * P_{sizePop}]$   
  while (!uvjetZaustavljanja) do  
    for ( $i = 1$  do  $sizePop$ ) do  
       $x \leftarrow P_i$   
       $x^{bazni} \leftarrow P^{bazni}_i$   
       $x \leftarrow istrazi(x, \Delta x_i)$   
      if ( $dobrota(x) > dobrota(x^{bazni})$ ) then  
         $y \leftarrow x$   
        for ( $j = 1$  do  $velicina(x)$ ) do  
           $x_j \leftarrow 2 * x_j - x^{bazni}_j$   
        end for  
         $x^{bazni} \leftarrow y$   
      else  
         $\Delta x \leftarrow \Delta x * \rho$   
         $x \leftarrow x^{bazni}$   
        if ( $\Delta x_i > \varepsilon \ \&\& \ dobrota(najbolji(P^{bazni})) > dobrota(x_i)$ ) then  
           $a \leftarrow selekcija(P, sizeP)$   
           $b \leftarrow selekcija(P, sizeP)$   
           $x \leftarrow krizanje(a, b)$   
           $x \leftarrow mutacija(x, p_m)$   
           $\Delta x_i \leftarrow \rho * x$   
           $x^{bazni} \leftarrow x$   
        end if  
      end if  
       $P^{bazni}_i \leftarrow x^{bazni}$   
       $P_i \leftarrow x$   
    end for  
  end while  
  return  $najbolji(P^{bazni})$   
end function
```

2.3.2. Genotip

Genotip u hibridnom algoritmu drugačiji je nego u genetskom algoritmu s lokalnom pretragom po tome što u genotipu sada imamo zapisane kompletne podatke o redoslijedu i parametrima filtera. Algoritam Hooke-Jeeves prilagođen je da radi i za cjelobrojne parametre kada ima informaciju o tome da li je neki parametar cjelobrojan ili ne. Genotip je definiran kao vektor:

$$q = [f_1 \ f_2 \dots f_{(broj\ filtera)} \ t \ p_1 \dots p_n]$$

$f_1 - f_{(broj\ filtera)}$ - parametri filtera u rasponu 0 - 1

t - prag kojim odlučujemo da li će se filter primjeniti ili ne u rasponu 0 - 0.5

$p_1 - p_n$ - svi parametri filtera

Parametri filtera određeni su realnim brojem u rasponu 0 - 1, a primjenjuju se samo ako im parametar pređe prag t . Redoslijed filtera određuje veličina parametra, tako da se oni s većom vrijednošću primjenjuju prvi. Uz genotip, definirani su i vektori s informacijom da li je parametar cjelobrojan, s minimalnom i maksimalnom vrijednošću parametara. Dodatni vektori koriste se kod generiranja jedinice djeteta, te određivanja koraka u Hooke-Jeeves algoritmu.

2.4. Funkcija dobrote

Funkcija dobrote jest funkcija kojom evaluiramo kvalitetu svakog rješenja. Što je vrijednost funkcije dobrote manja rješenje je bolje. Funkcija dobrote obrađuje sliku filterima opisanim u rješenju iz čega se dobiva matrica ispunjena vrijednostima 0 i 1. Jedinice znače da je to područje čvorića, dok nule označavaju područje bez čvorića.

Baza slika pluća JSRT za svaku RTG snimku ima masku koja određuje gdje je čvorić. Uspoređivanjem naše dobivene maske i maske s točnom lokacijom čvorića dobivamo dva podatka, broj lažno pozitivnih i broj lažno negativnih piksela. Važno nam je razlikovati ta dva podatka jer ih ne kažnjavamo jednako. Dobrota se računa prema sljedećoj formuli :

$$fitness = FP + FN * e^{(10 * (FN/P - alpha))}$$

FN - broj lažno negativnih piksela

FP - broj lažno pozitivnih piksela

P - broj pozitivnih piksela

$alpha$ - parametar koji određuje postotak otkrivenih pozitivnih piksela nakon kojeg

algoritam počinje minimizirati lažno pozitivne

Kada je postotak otkrivenih pozitivnih piksela manji od *alpha*, algoritam više kažnjava lažno negativne piksele, tj. nastoji minimizirati njihov broj. Kada je postotak otkrivenih pozitivnih piksela veći od *alpha*, algoritam više kažnjava lažno pozitivne, minimizira broj lažno pozitivnih. Konstanta *alpha* je iskustvena, te ovisi o problemu.

2.5. Opis korištenih filtera

Filteri koje koristimo za obradu slike imaju velik utjecaj na dobrotu krajnjeg rješenja. Bilo bi dobro što više proširiti skup filtera koje koristimo što raznovrsnijim filterima. No, s povećanjem broja parametara koje optimiziramo dolazi do značajnog usporavanja algoritma. Treba naći ravnotežu između broja filtera i brzine izvođenja. Filteri korišteni u ovom radu su:

1. **Morfološki filteri** - kao parametar primaju strukturni element. Strukturni element je matrica koja se sastoji od nula i jedinica, a predstavlja lik nad kojim se primjenjuje filter.
 - **Erozija** - sažima svjetle objekte na slici, proširuje tamna područja. Strukturni element pomiče se po slici, cijelom području prekrivenim strukturnim elementom dodjeljuje se svjetlina najmanje svjetlog piksela iz područja prekrivenog strukturnim elementom. Utjecaj filtera na sliku prikazan je na slici 2.1f.
 - **Dilatacija** - proširuje svjetla područja. Radi slično kao i erozija, osim što području prekrivenim strukturnim elementom dodjeljuje svjetlinu najsvjetlijeg piksela u području. Utjecaj filtera na sliku prikazan je na slici 2.1e.
 - **Zatvaranje** - filtrira sliku tako da nad njim prvo nad njom primjeni eroziju, a zatim dilataciju. Zatvara tamne pukotine na slici. Utjecaj filtera na sliku prikazan je na slici 2.1c.
 - **Otvaranje** - sličan filtru zatvaranja. Nad slikom primjenjuje dilataciju, pa zatim eroziju. Uklanja male objekte sa slike. Utjecaj filtera na sliku prikazan je na slici 2.1b.
 - **Top hat** - oduzima filterom otvaranja obrađenu sliku od originalne. Izdvaja manje objekte na slici. Utjecaj filtera na sliku prikazan je na slici 2.1d.



Slika 2.1: Morfološki filteri

2. Filteri za prilagođavanje kontrasta

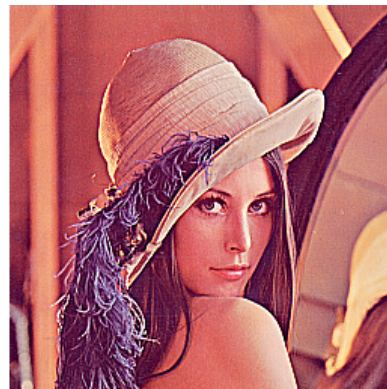
- **Izoštavanje** - izoštrava sliku uz pomoć unsharp filtera. Slika se prvo zamuti unsharp filtrom, te se oduzima od originalne slike. Utjecaj filtera na sliku prikazan je na slici 2.2b.
- **Gaussovo zamućivanje** (*eng. Gaussian blur*) - zamućuje sliku. Nisko propusan filter u obliku Gaussove funkcije. Za parametar prima stan-

dardnu devijaciju Gaussove razdiobe. Utjecaj filtera na sliku prikazan je na slici 2.2a.

- **Ujednačavanje histograma** - povećava kontrast slike. Ujednačava distribuciju piksela prema svjetlini koju poprimaju. Utjecaj filtera na sliku prikazan je na slici 2.2e.
- **Prilagođavanje intenziteta** (*eng. intensity value adjustment*) - povećava kontrast filtrirane slike. Djeluje tako da 1% elemenata slike postavi na vrijednosti visokog ili niskog intenziteta. Elemente velike svjetline posvjetljuje, a tamne elemente potamnjuje. Vrijednost visokog i niskog intenziteta se određuje za svaku sliku. Utjecaj filtera na sliku prikazan je na slici 2.2d.



(a) Gaussovo zamućivanje



(b) Izoštavanje



(c) Originalna slika, crno -
bijela



(d) Prilagođavanje intenziteta

(e) Ujednačavanje histograma

Slika 2.2: Filteri za prilagođavanje kontrasta

3. **Određivanje praga (eng. *threshold*)** - slika se preslikava u crno-bijelu tako da se svi pikseli svjetline veće od praga bojaju bijelo, a ostali crno. Ovaj filter koristi se na kraju kako bi se dobila matrica u kojoj jedinice označuju položaj čvorića, a nule područje bez čvorića. Utjecaj filtera na sliku prikazan je na slici 2.3.



Slika 2.3: Filter praga

2.6. Korištena baza podataka

Koristi se digitalna baza RTG slika koju je izdalo Japansko radiološko društvo (eng. Japanese Society of Radiological Technology - JSRT). Baza se sastoji od 247 rendgenskih snimki pluća, sa i bez čvorića. Također, baza sadrži primjere i benignih i malignih čvorića. Svaka snimka iz baze ima svoju masku koja pokazuje položaj čvorića. Za ispitivanje algoritama korišten je skup od 3 RTG snimke. Korištene su slike

dimenzija 256x256 piksela. Za svaku snimku napravljene su maske koje odvajaju područja plućnih krila od pozadine, te je napravljeno ujednačavanje svjetline svake slike na prosječnu svjetlinu cijelog skupa.

2.7. MATLAB

Za implementaciju algoritama korišten je programski jezik Matlab s Matlab razvojnom okolinom. Matlab je izabran zbog optimizacija u radu s matricama, te mogućnostima paralelizacije. Korištene su Matlab biblioteke Global Optimization Toolbox i Parallel Computing Toolbox. General Optimization Toolbox sadrži ugrađene funkcije za optimizaciju genetskim algoritmom i simpleks postupkom po Nedleru i Meadu koje su korištene u radu. Iz Parallel Computing Toolbox-a korištena je mogućnost paralelizacije pokretanjem na više procesa naredbom `matlabpool`. Naredba `matlabpool` stvara zadani broj Matlab procesa radnika, koji pomažu glavnoj kopiji Matlab-a u računanju.

3. Rezultati

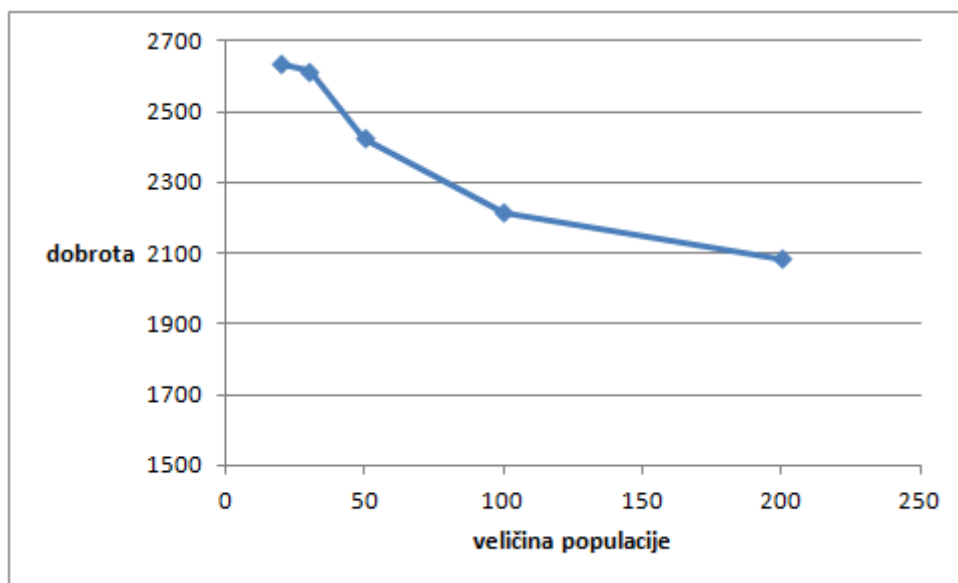
Ispitivanje algoritama provodilo se tako da je algoritam pokretan više puta sa različitim iznosima ulaznog parametra, nakon čega se taj parametar fiksirao, a ispitivanje nastavilo za druge ulazne parametre. Iznos ulaznog parametra iz ispitnog skupa odabiran je na osnovu dobrote i vremena izvođenja rješenja dobivenih kao prosjek 10 izvođenja. Ispitivanje je provedeno na skupu od 3 RTG snimke. Nakon provedenog ispitivanja, rezultat se dobiva primjenom najboljeg rješenja na sve slike iz baze.

3.1. Genetski algoritam s lokalnom pretragom

Ispitivanje genetskog algoritma s lokalnom pretragom sastojalo se od namještanja parametara algoritma, ispitivanja algoritama lokalne pretrage i ispitivanja dva različita genotipa. Za namještanje veličine populacije, uvjeta zaustavljanja i broja generacija GA u iteraciji algoritma korišten je algoritam Hooke-Jeeves kao algoritam lokalne pretrage i genotip kao realni vektor koji određuje redoslijed i parametre filtera.

3.1.1. Utjecaj veličine populacije

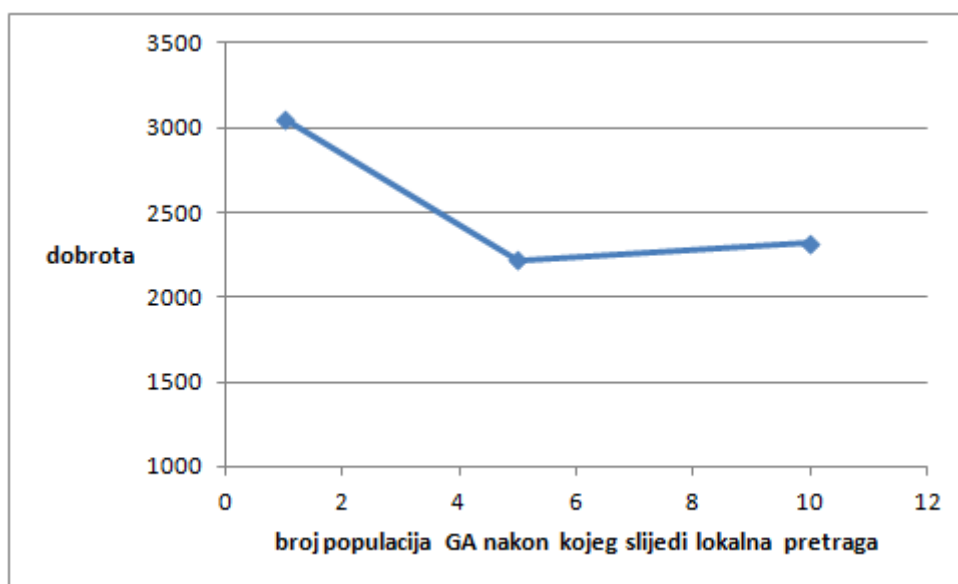
Algoritam je ispitan na populacijama od 20, 30, 50, 100 i 200 jedinki, s pokretanjem lokalne pretrage svakih 5 generacija genetskog algoritma. Algoritam se zaustavljao kada 5 uzastopnih iteracija ne nađe bolje rješenje. Rezultat ispitivanja vidi se na slici 3.1. Za nastavak ispitivanja odabrana je veličina populacije 100 zbog dobrih rezultata i znatno kraćeg vremena izvođenja nego s populacijom veličine 200 jedinki.



Slika 3.1: Utjecaj veličine populacije

3.1.2. Nakon koliko generacija GA pokrenuti lokalnu prtragu

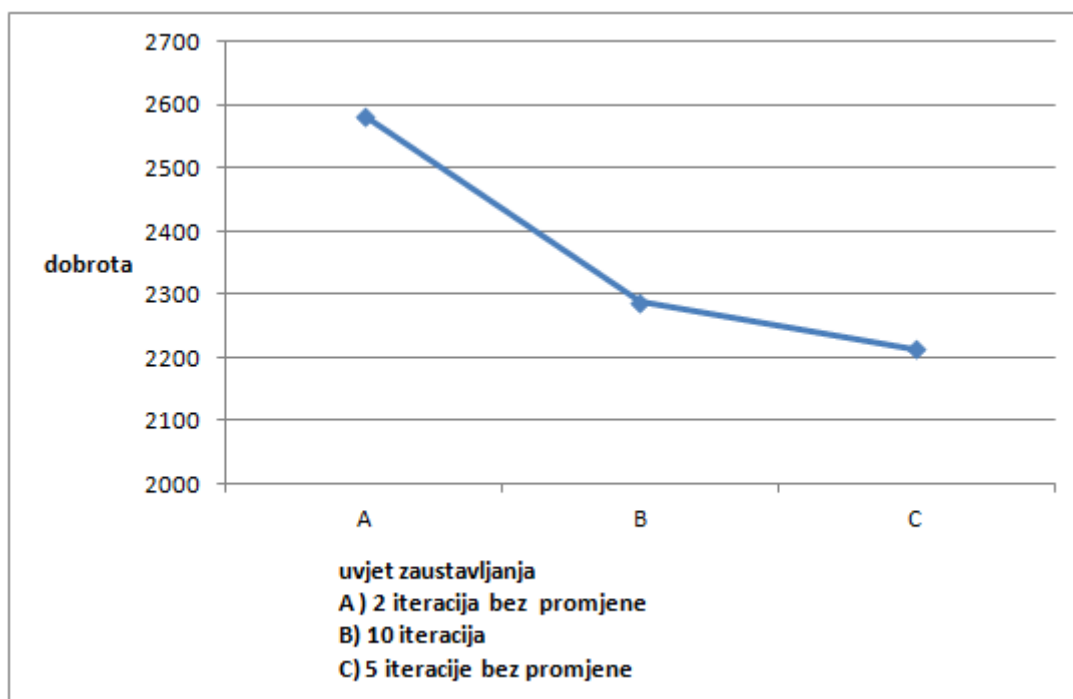
Algoritam je ispitan s lokalnom pretragom svakih 1,5 i 10 generacija genetskog algoritma. Veličina populacije bila je 100 jedinki, a uvjet zaustavljanja 5 iteracija algoritma bez nalaska boljeg rješenja. Rezultat ispitivanja vidi se na slici 3.2. Veći broj generacija GA nije se pokazao boljim, a vrijeme potrebno za izvođenje je znatno duže. Izabrano je pokretanje lokalne pretrage svakih 5 generacija genetskog algoritma.



Slika 3.2: Nakon koliko generacija GA pokrenuti lokalnu pretragu

3.1.3. Uvjet zaustavljanja

Algoritam je ispitan na 3 uvjeta zaustavljanja, to su: 5 iteracija algoritma bez poboljšanja najbolje dobrote, 2 iteracije bez poboljšanja dobrote, te fiksno 10 iteracija. Ispitivanje je provedeno s populacijom veličine 100 jedinki i lokalnom pretragom svakih 10 generacija GA. Rezultat ispitivanja vidi se na slici 3.3. Za daljnje ispitivanje odabran je fiksni broj iteracija zbog kraćeg vremena izvođenja i samo malo lošijeg rezultata od zaustavljanja nakon 5 iteracija bez promjene dobrote.



Slika 3.3: Uvjet zaustavljanja

3.1.4. Algoritmi lokalne pretrage

Algoritam je ispitan uz dva različita algoritma lokalne pretrage. GA s Hooke-Jeeves algoritmom lokalne pretrage ispitan je uz populaciju veličine 100 jedinki, pokretanjem lokalne pretrage svakih 5 generacija genetskog algoritma i zaustavljanjem nakon 10 iteracija algoritma. Lokalna pretraga zaustavlja se nakon 5 iteracija Hooke-Jeeves algoritma. GA sa Simpleks algoritmom lokalne pretrage ispitan je uz istu veličinu populacije, uvjet zaustavljanja cijelog algoritma, te broj generacija GA nakon kojeg se pokreće lokalna pretraga, ali s konvergencijom rješenja kao uvjetom zaustavljanja lokalne pretrage. U tablici 3.1 dani su prosjeci najboljih rješenja iz 10 pokretanja oba algoritama. Hooke-Jeeves algoritam nešto je bolji za pronalazak optimalnog rješenja ovog problema, no razlika među algoritmima nije velika.

Tablica 3.1: Algoritmi lokalne pretrage

Algoritam	Dobrota
Hooke-Jeeves	2288.36
Simpleks postupak	2393.51

3.1.5. Utjecaj genotipa

Ispitivanje algoritma provedeno je uz dva različita genotipa jedinki. Redoslijed primjene filtera u jednom je genotipu prikazan slijedom realnih brojeva čija veličina utječe na odabir i redoslijed filtera koji se primjenjuju, dok drugi sadrži slijed cijelih brojeva koji označavaju redoslijed primjene filtera. U tablici 3.2 prikazani su prosjeci najboljih rješenja iz 10 pokretanja algoritma za oba genotipa. Prikaz redoslijeda slijedom realnih brojeva daje mnogo bolja rješenja.

Tablica 3.2: Utjecaj genotipa

Prikaz genotipa	Dobrota
Slijed cijelih brojeva	3578.37
Slijed realnih brojeva	2288.36

3.2. Hibridni algoritam

Ispitivanje hibridnog algoritma sastojalo se od namještanja veličine populacije korištene u algoritmu. Uvjet zaustavljanja bile su 4 iteracije bez promjene najbolje dobrote ili ukupno 35 iteracija algoritma, do čega dođe prvo. Uvjet konvergencije Hook-Jeeves algoritma bio je korak pretraživanja manji od 0.01, dok se korak smanjivao na polovinu iznosa nakon svake iteracije. Hibridni Hooke-Jeeves algoritam polučio je dobre rezultate, ali uz veliku vremensku cijenu. Konvergencija prema optimalnom rješenju spora je, te bi algoritam trebalo ispitati uz dulje vrijeme izvođenja.

3.2.1. Utjecaj veličine populacije

Algoritam je ispitan na populacijama od 20 i 30 jedinki. U tablici 3.3 prikazani su prosjeci najboljih rješenja iz 10 pokretanja algoritma za obje veličine populacije. Iz tablice vidi se da dobrota rješenja raste s veličinom populacije, te bi algoritam trebalo

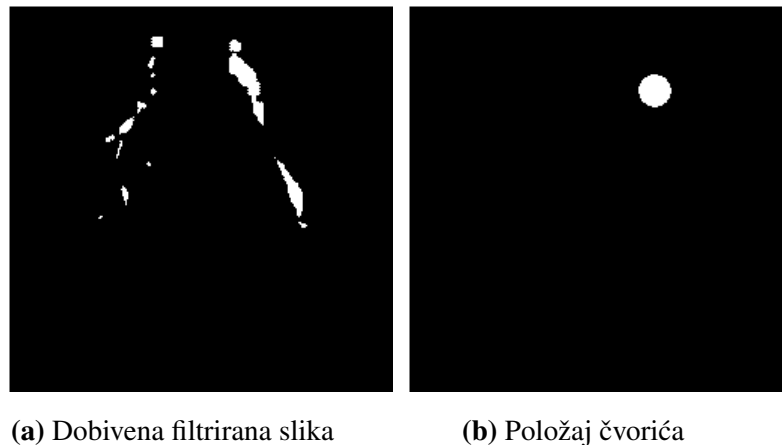
ispitati i na populacijama veće veličine. Također, povećanje populacije znatno utječe na vrijeme izvođenja.

Tablica 3.3: Utjecaj veličine populacije

Veličina populacije	Dobrota
20	2165.70
30	1986.30

3.3. Ispitivanje najboljeg rješenja

Najbolje rješenje nađeno je hibridnim Hooke-Jeeves algoritmom uz veličinu populacije od 20 jedinki nakon 63 iteracije algoritma. Prosječna dobrota najboljeg nađenog rješenja jest 1590.45 na skupu za učenje, te 5185.4 na cijeloj bazi. Slika 3.4a prikazuje najbolje filtriranu sliku ovog rješenja koja sadrži čvorić, te ima dobrotu 939.55.



Slika 3.4: Rješenje dobrote 939.55

Ispitivanjem hibridnog Hooke-Jeeves algoritma s veličinom populacije 30 jedinki nađeno je još jedno veoma dobro rješenje. Prosječna dobrota rješenja jest 1674.4 na skupu za učenje, te 5637.1 na cijeloj bazi podataka. Slika 3.5a prikazuje najbolje filtriranu sliku ovog rješenja, ali i cijelog ispitivanja. Slika 3.6a prikazuje najbolje filtriranu sliku koja nije dio skupa za učenje.



(a) Dobivena filtrirana slika

(b) Položaj čvorića

Slika 3.5: Najbolje rješenje

dobrota = 695.91



(a) Dobivena filtrirana slika

(b) Položaj čvorića

Slika 3.6: Rješenje dobrote 852.05

4. Zaključak

Dijagnoza malignih tvorevina na RTG slikama pluća problem je pogodan za rješavanje računalom metodama umjetne inteligencije zbog svojstva plućnih čvorića da nemaju tipičan uzorak, te da ne postoji metoda kojim možemo sa sigurnošću zaključiti da li su maligni ili benigni. Pouzdan alat za obradu RTG slika pluća omogućio bi detekciju malignih tvorevina u ranoj fazi bolesti, te time i rani pristup liječenju i veću mogućnost izlječenja. Ovaj rad usmjeren je na detekciju čvorića na RTG slikama pluća dok je određivanje da li su čvorići maligni ili benigni ostavljeno za daljnja istraživanja.

Slika pluća obrađena je filterima kako bi se izdvojila poručja čvorića od ostatka pluća. Genetski algoritam s lokalnom pretragom i hibridni algoritam korišteni su za optimiranje redoslijeda i parametara filtera koji se primjenjuju na slici, dok su kao algoritmi lokalne pretrage korišteni Hooke-Jeeves algoritam i Simpleks algoritam po Nelderu i Meadu. GA s lokalnom pretragom ispitan je na populacijama različite veličine, s različitim uvjetima zaustavljanja, genotipima i brojem iteracija genetskog algoritma nakon kojeg slijedi lokalna pretraga. Hibridni algoritam ispitan je na populacijama različitih veličina, te je dao veoma dobre rezultate, no uz veliku vremensku složenost učenja algoritma.

Poboljšanja algoritama mogla bi se ostvariti dodavanjem novih filtera u genotip, no potreban je oprez zbog povećanja vremena izvođenja. Također, hibridni algoritam pokazao se sporim u konvergenciji prema optimalnom rješenju, te bi dulje izvođenje dalo bolje rezultate, ali uz vremenski trošak. Iako učenje algoritama traje dugo, primjena filtera dobivenih algoritmom je trenutna, te bi korištenje ovog alata kod dijagnostike bilo dovoljno brzo.

LITERATURA

- [1] Hiroyuki Abe Heber MacMahon Kunio Doi Kenji Suzuki, Junji Shiraishi. False-positive reduction in computer-aided diagnostic scheme for detecting nodules in chest radiographs by means of massive training artificial neural network. *Academic Radiology*, 12:191–201, 2005.
- [2] Danko Komlen. Modeliranje genske regulacijske mreže pomoću hibridnog koevolucijskog algoritma. Magistarski rad, Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva, 2012.
- [3] Riccardo Poli. Genetic programming for feature detection and image segmentation. U *University of Sussex, UK*, stranice 110–125. Springer-Verlag, 1996.

Detekcija malignih tvorevina na RTG slikama pluća uz pomoć genetskog algoritma i determinističkih algoritama kontinuirane optimizacije

Sažetak

U ovom radu dan je opis problema detekcije malignih tvorevina na RTG slikama pluća, koristi od rješavanja problema uz pomoć računala, te mogući daljnji smjerovi u rješavanju metodama umjetne inteligencije. Detekcija čvorića na plućima u radu provodi se primjenom filtera na RTG sliku pluća, pri čemu je genetski algoritam sa lokalnom pretragom korišten za optimizaciju redoslijeda i parametara filtera koji se primjenjuju. Ispitana je korist od rješavanja ovog problema hibridnim algoritmom, te genetskim algoritmom uz lokalnu pretragu Hooke-Jeeves i Simpleks algoritmom po Nedleru i Meadu. Ispitivanje je provedeno uz namještanje parametara navedenih algoritama, te su dani rezultati ispitivanja pojedinih algoritama.

Ključne riječi: Genetski algoritam, GA, Hooke-Jeeves algoritam, Simpleks algoritam po Nedleru i Meadu, hibridni algoritam

Title

Abstract

In this paper, the problematics of detection of malicious nodules on RTG lung images, solving benefits and possible solving approaches are described. Detection of malicious nodules is conducted by filtering the image. Filter sequence and parameters are defined with the help of genetic algorithm with local search and hybrid algorithm. Local search algorithms implemented are Hooke-Jeeves algorithm and Nedler-Mead method. Results of testing are presented and adjusting of algorithm parameters is conducted.

Keywords: Genetic algorithm, GA, Hooke-Jeeves algorithm, Nedler-Mead method, hybrid algorithm